Phylogenetic approaches for detecting fragmentation in genome and transcriptome annotations

Ivana Pilizota

UCL

Department of Genetics, Evolution and Environment

2020

Thesis submitted to the UCL for the degree of Doctor of Philosophy

Declaration

I, Ivana Pilizota, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis. Specifically, the chapters of the thesis involve joint work which is specified below:

Chapter 2. This chapter is a modification of a joint work with Lucas D. Wittwer, Adrian M. Altenhoff and Christophe Dessimoz, of which I am co-first author, and published in Wittwer *et al.* (2014). I compared the proposed method to the chosen *k*-mer approaches and wrote the paper. In addition, two paragraphs from the Introduction section of the paper were modified and included in **section 1.4.3** in **Chapter 1: Introduction**.

Chapter 3. A modification of this chapter has been published in Piližota *et al.* (2019) which I wrote as a first author. Co-author Adrian Altenhoff helped me with turning my scripts into a stand-alone tool ESPRIT 2. (Clément-Marie Train performed an analysis which does not form part of the thesis.) Some content from Introduction of the paper was extended to form **section 1.4.4** in **Chapter 1: Introduction**, while the first two sentences of Abstract appear in section **1.5 Research questions and overview of the thesis**, and the third sentence was included in its modified form.

Chapter 6. This chapter contains parts of Abstracts from Wittwer *et al.* (2014) and Piližota *et al.* (2019).

In addition to the above, during my PhD studies I co-authored the following paper, which does not form part of the thesis:
Altenhoff, A. M. *et al.* (2014) 'The OMA orthology database in 2015: function predictions, better plant support, synteny view and other improvements', *Nucleic Acids Research*, 43(Database issue), pp. D240–D249.

Abstract

The landscape of biological research and innovation has been transformed with the invention of genome sequencing methods and corresponding assembly and annotation algorithms. Yet many assemblies and annotations remain fragmented limiting applications which require more complete and reliable datasets.

The goal of this thesis was to establish methods to detect fragmentation in genome and transcriptome annotation by exploiting available data from related species in a phylogenetic framework.

Prior to applying core methods to detect fragmentation, it is important to establish informative sequences from related species, i.e. putative homologs. This typically requires all-against-all protein-protein sequence comparison within and across species in the dataset. To speed up this process, we developed an approach which attempts to incorporate transitive property of homology and considers putative homology on putative protein subsequences.

Putative homologs can then be used as input for our phylogenetic heuristics to detect fragments of the same gene model in the genome assembly of interest. One heuristic collapses internal tree branches with low SH-like branch support, the other exploits a likelihood ratio value. The heuristics found 1,221 pairs of distinct gene models in the challenging putative bread wheat genome which we believe are actually fragments of the same gene model.

We also employed the heuristics on the putative genome of wild olive and identified 102 pairs of distinct gene models, potentially fragments of the same model. Importantly, we provide guidelines on assessing predictions based on the data at hand.

Finally, we started exploring behaviour of the heuristics on the transcript models constructed on the cassava transcriptome assembly. Due to time constraints, the outcomes of the study are limited but hopefully provide sound guidelines for further work.

The methods are not restricted to the plant kingdom and can already be used on any species in their current state.

Impact statement

Genome and transcriptome sequencing have transformed biological research across all domains of life. Sequencing is often the first step, followed by: 1) assembling, the process of reconstructing typically unknown DNA or RNA sequences, and 2) annotation, the process of identifying the newly constructed sequences and possibly ascribing them function. Despite scientific and technological advances over the past decades, assembling remains a difficult task yielding fragmented assemblies which can also contain misassembled sequences and uncorrected sequencing errors. Subsequently, the annotation pipeline, incapable of dealing with all assembly artefacts and imperfections, assigns gene models which actually represent only fragments of the true genes. This then affects downstream analyses which require complete and accurate datasets.

The challenges of sequencing, assembling and annotation increase with the increase in complexity of the genome under study. The problems are particularly pronounced in plants, arising from their complex evolutionary histories. Yet, plant data is crucial for the agricultural and biotechnological innovations necessary to meet the ever-growing demands for food.

Our work introduces innovative methodology to improve fragmented genome annotation in a phylogenetic setting. We developed two phylogenetic heuristics which detect gene models capturing different parts of the same gene, based on information from a reference set of gene models derived from putative evolutionarily-related genes across closely related species. We applied the methods to the challenging fragmented putative genome of bread wheat and to the only available genome assembly and annotation of wild olive to date. The predictions could be further investigated to determine the cause of fragmented annotation. In particular, if the fragmentation already existed in the corresponding assembly, the predictions could aid improving genome assemblies as well.

To allow for the application of methods on a researcher's dataset of interest, we provide their source code which can be used on any species, including outside the plant kingdom (https://github.com/DessimozLab/esprit2). We also provide a step-by-step guide which can help to assess the methods' behaviour on a dataset and the predictions they make without using any additional data, which will often be the case for a newly sequenced species. In addition, we provide suggestions for further research on adapting the heuristics to the transcriptome annotation.

The heuristics require reference gene models of putative homologs from other species. Yet homology inference can be a computational bottleneck in a pipeline. We thus developed an algorithm which attempts to speed up the inference by considering transitive property of homology and subsequence-level homology. Its source code is also publicly available.

In the era of accumulating data, we hope that our approaches will aid scientific research and motivate further applications, refinements and method developments. With ongoing and future sequencing projects, the number of available annotated genome and transcriptome assemblies will continue to increase. Importantly, this will also reduce the average evolutionary distance between the represented species, and facilitate further applications and advancements in the field of comparative genomics. Hopefully, this work makes a contribution towards exploiting the full potential of sequence data.

Acknowledgements

“It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way...”

Charles Dickens, *A Tale of Two Cities*

First and foremost, I would like to thank my supervisors Dr Christophe Dessimoz and Dr Henning Redestig for their immense support, encouragement and guidance with a healthy dose of trust to let me pull myself up by my bootstraps.

I truly appreciate the kindness of my colleagues in Ghent, Lausanne, London and Zurich who enriched my experience as a PhD student both professionally and personally.

I am deeply thankful to my family and friends for being part of this journey and helping me to keep moving forward even when I could not walk.

Last but not least, I would like to thank Prof Richard Goldstein and Dr Matthieu Muffato for their involvement and revealing to me the distance I can go.

Thank you most profoundly,
Ivana Piližota

Table of contents

List of figures

List of tables

Chapter 1: Introduction

1.1 Genome and transcriptome assemblies

The ability to include genomic (DNA) and transcriptomic (RNA) data into biological studies has accelerated research across all domains of life. It has been facilitated with the development of sequencing technologies and assembling approaches which attempt to reconstruct DNA or RNA sequences from the sequence fragments. Being faced with the difficult task of putting back the pieces into an often unknown and large jigsaw puzzle, assemblers cannot resolve all the issues and hence produce fragmented assemblies with various mistakes within the fragments.

1.1.1 Sequencing

In 1980, the Nobel Prize in Chemistry was awarded in part to Walter Gilbert and Frederick Sanger "for their contributions concerning the determination of base sequences in nucleic acids" (Kolata, 1980)—their independent groundbreaking work from the 1970s (Sanger and Coulson, 1975; Maxam and Gilbert, 1977; Sanger, Nicklen and Coulson, 1977) which opened the door to genome and transcriptome sequencing, and initiated the revolution in biological sciences.

The Sanger method (Sanger, Nicklen and Coulson, 1977) prevailed in industry for decades. In their experiments, Sanger, Nicklen and Coulson managed to determine 15-200 nucleotides of a sequence, sometimes even up to 300 “with reasonable accuracy”. Over the years, the method was further improved leading to today’s automated high throughput Sanger sequencing methods which can provide ~1,000 bp long reads with up to 99.999% accuracy (Shendure and Ji, 2008).

The next-generation sequencing (NGS), high-throughput sequencing (HTS) or the second-generation sequencing technologies started another revolution as they made sequencing affordable to individual laboratories at a fraction of the Sanger sequencing costs (Shendure and Ji, 2008; Metzker, 2009; Goodwin, McPherson and McCombie, 2016). Its invention and commercialisation in mid 2000s caused the exponential increase of the number of available putative genomes, fostered population genomics studies, enriched the medical data and advanced biotechnological innovation (Shaffer, 2007; Zynda, 2014). Some of the drawbacks coming from shorter (35-700 bp) and more error-prone (~0.1-15%) reads can be overcome with higher sequencing depth, i.e. by sequencing target genome or target genomic region multiple times (e.g. 10x or even 10,000x) (Goodwin, McPherson and McCombie, 2016).

The third-generation sequencing methods emerged around the year 2010 as a potential solution to the limitations of the NGS reads—reconstruction of challenging genomic regions and fragmentation in assemblies caused by short reads (McCarthy, 2010). They are able to provide longer reads but with relatively high error rates and higher costs than the NGS technologies (Sedlazeck *et al.*, 2018; Wang *et al.*, 2019). For instance, a 2015 review of the Pacific Biosciences (PacBio) sequencing platform reported a median read length of around 20 kbp but with error rates between 11 and 15% (Rhoads and Au, 2015). Publications document very long (ultra-long) reads generated by Oxford Nanopore Technologies (ONT) of up to ~1Mb and error rates of 12-38% (Besser *et al.*, 2018; Jain *et al.*, 2018; van Dijk *et al.*, 2018)[1]. Consequently, computational and experimental approaches for error correction have been developed (Laehnemann, Borkhardt and McHardy, 2016; van Dijk *et al.*, 2018). Computational methods attempt to detect and trim, filter out or correct affected reads, usually relying on the information already contained in the dataset (e.g. per base quality scores, base frequencies) but they can also incorporate additional data such as short

[1] Some literature classifies nanopore-based sequencing as the fourth-generation sequencing technology, e.g. Feng *et al.* (2015), Bhadauria (2017).

reads[2]. A new type of reads—linked-reads—has been developed by 10X Genomics (Zheng *et al.*, 2016). They consist of short Illumina reads linked together sparsely covering 100 kbp on average (Zheng *et al.*, 2016; Sedlazeck *et al.*, 2018). Short NGS reads ensure high read accuracy, the sparsity reduces the costs of sequencing and linking provides long-range information (Zheng *et al.*, 2016).

Having obtained a set of raw sequencing reads, the next step is a genome or transcriptome assembly.

1.1.2 Assembling

Starting with sequencing reads, there are two main types of sequence reconstruction: genome assembly and transcriptome assembly. The aim of genome assembly is to reconstruct the putative genome sequence and obtain one gapless sequence per chromosome representing all genes and intergenic regions (Baker, 2012). Transcriptome assembly attempts to capture all expressed transcripts and their isoforms represented by the reads (Martin and Wang, 2011)[3]. A high-quality genome assembly aids studying genome structure, gene function, evolution of genes and species, discovering genetic variation and associated biological traits within and across species (Hunt *et al.*, 2014; Chaisson, Wilson and Eichler, 2015; Muggli *et al.*, 2015; Meltz Steinberg *et al.*, 2017). While a genome assembly provides a static picture of a genome, transcriptome assembly captures what is going on in a certain cell (tissue or organism) under certain conditions at a certain developmental stage (Wang, Gerstein and Snyder, 2009; Manzoni *et al.*, 2018). Furthermore, sequencing and assembling a transcriptome is typically less complex and cheaper than it is for a genome (Góngora-Castillo and Buell, 2013). Thus, in the absence of a (high-quality) genome assembly,

[2] An extensive list of error correction tools as well as those implementing this task as one of the steps in a more complex pipeline can be found in, e.g. Laehnemann, Borkhardt and McHardy, 2016.

[3] A historical overview of the term “gene” (coupled with key findings on gene transcription which had an effect on the definition of a gene) can be found in, e.g. Gerstein *et al.* (2007) and Portin and Wilkins (2017).

researchers often opt for a transcriptome assembly if it provides adequate information for their purposes (Góngora-Castillo and Buell, 2013). A transcriptome assembly can also reveal previously unknown genes and improve existing genome annotation (Lu, Zeng and Shi, 2013)[4].

There are three main differences between genome and transcriptome reads which affect their assembly (Martin and Wang, 2011). First, higher sequencing depth reveals repetitive regions in a genome whereas it indicates abundance in transcriptomes. Second, in genome sequencing projects, both strands are sequenced while that may not be the case with transcriptomes. Third, isoforms of the same gene which share exons pose difficulties in transcriptome assemblies. However, both genome and transcriptome assembly exploit the same strategies with modifications depending on sequencing technology and peculiarities of the input data. In fact, numerous assembly tools were initially developed for genomic data and later extended for transcriptome input.

There are also differences between reads obtained by different sequencing technologies, such as read lengths and sequencing error rates (see section 1.1.1), which have to be accommodated by assemblers and assembly pipelines. In the rest of the section, we describe assembling the second-generation sequencing reads and briefly refer to the third-generation reads assembly. Conceptually, the assembly process is the same regardless of the technology: reads are extended to contigs, and in the case of the genome assemblies contigs are further ordered, oriented and linked into scaffolds which are ordered and linked into putative chromosomes.

For a newly sequenced species, the assembly process starts with identifying gapless overlapping reads and forming contiguous sequences—contigs (Baker, 2012). The existing algorithms are referred to as *de novo* algorithms, and typically fall into one of three categories: the greedy approach, the overlap layout consensus approach and the *de Bruijn* graph approach. More

[4] More about genome and transcriptome annotation in section 1.2

details on the approaches and tools can be found in, e.g. Zhou *et al.* (2002), Pop (2009), Li *et al.* (2012), Bradnam *et al.* (2013), Nagarajan and Pop (2013), Wang and Gribskov (2017), Voshall and Moriyama (2018). Regardless of the approach, extending sequences based on overlap is not trivial.

If the assembly of sequenced species or very closely related species is available, reads can be assembled using a reference-based assembly (Lee and Tang, 2012). By mapping the reads to the reference first and then assembling them, contamination and sequencing artefacts are easier to detect as they will unlikely align to the reference. A genome assembly can also be used as a reference in the process of transcriptome assembly (Voshall and Moriyama, 2018). If a genome assembly is not sufficiently similar, a putative proteome can serve the same purpose. Examples of tools employed in reference-based assembling can be found in, e.g. Schneeberger *et al.* (2011), Steijger *et al.* (2013), Voshall and Moriyama (2018).

A reference-based approach can be combined with a *de novo* approach (Sohn and Nam, 2016). Following a reference-based assembly, contigs can be assembled with unmapped reads using a *de novo* algorithm. *De novo* assembly can also be computed for each loci separately taking the advantage of parallel computing. If the available reference assembly is of low quality or from a closely related species, the first step can be to do *de novo* assembly followed by mapping resulting contigs and unassembled reads to the reference assembly in order to join contigs and fill in the gaps between them (Silva *et al.*, 2013). This way the dependence on the reference assembly is weaker and errors in the reference may not propagate to the target assembly. Furthermore, novel and diverged genes can be assembled *de novo*. This is particularly important to transcriptome data for novel and trans-spliced transcripts, as well as for transcripts missing in the genome assembly. Less conserved non-coding regions will also benefit from *de novo* assembling.

In genome assembly projects, contig assembly is typically followed by linking contigs into scaffolds (Hunt *et al.*, 2014; Ghurye *et al.*, 2017). The ultimate goal of scaffolding is one sequence per chromosome which might contain gaps of correctly estimated lengths. Some contig assemblers already incorporate a scaffolding module. However, scaffolding is usually performed as a separate step of the assembly pipeline and uses complementary information derived from multiple sequencing technologies or experiments to improve the contiguity of the assembly. Scaffolders typically incorporate paired-end reads and mate pair libraries, and information available from the sequencing experiment such as ordering, orientation and the expected distance between the reads. In addition, RNA-seq data, long reads, chromatin interaction datasets and optical mapping can be employed for scaffolding or even placing scaffolds within chromosomes (Mortazavi *et al.*, 2010; Burton *et al.*, 2013; Dong *et al.*, 2013; Rice and Green, 2019). This kind of linearisation of contigs into scaffolds is not applicable to transcriptome assemblies due to multiple transcript isoforms being expressed by a single gene (Xie *et al.*, 2014). Extensive evaluations of scaffolding tools can be found in, e.g. Hunt *et al.* (2014), Mandric, Knyazev and Zelikovsky (2018).

Once the contigs are ordered and oriented into scaffolds, it is easier to reconstruct the sequences in the gaps, if desired. This process is often called gap filling, gap closing or genome finishing (Boetzer and Pirovano, 2012; Paulino *et al.*, 2015; Ghurye and Pop, 2019; Rice and Green, 2019). Gaps can be filled by identifying and assembling initial sequencing reads, resequencing the target regions or using additional reads from a different technology (Kremer, McBride and Pinto, 2017; Ghurye and Pop, 2019). An assembly built by combining sequencing data from different technologies is often referred to as hybrid assembly (Utturkar *et al.*, 2014; De Maio *et al.*, 2019).

Assembling third-generation sequencing reads requires algorithms capable of dealing with long read lengths and high error rates (Sedlazeck *et al.*, 2018). Typically, the assembling process starts with read error correction either using NGS reads (for low long-read coverage, i.e. < 30x) or based on

self-correction, i.e. exploiting the alignment of long reads to themselves which tends to be more successful but also more expensive as it requires higher levels of coverage (Sedlazeck *et al.*, 2018; Jung *et al.*, 2019). Corrected reads are then assembled into contigs with algorithms accommodating long reads and tolerating remaining errors within them. In the following step—contig consensus polishing—errors within contigs are corrected using paired-end or mate pair NGS reads (limited to repetitive sequences but generally leads to higher accuracy) or the reads used for assembly (Sedlazeck *et al.*, 2018; Jung *et al.*, 2019). Despite high error rates in the reads, the estimated assembly error rates can be lower than 1% (Sedlazeck *et al.*, 2018). Genome assembling then continues with scaffolding. The best results have been reported with employing BioNano Genomics optical mapping, Hi-C data and 10X Genomics linked-reads (Sedlazeck *et al.*, 2018). That type of data can also help ordering and linking scaffolds into putative chromosomes (Jung *et al.*, 2019). Additional long reads and linked-reads can help gap filling (Sedlazeck *et al.*, 2018; Jung *et al.*, 2019). Sedlazeck *et al.*, 2018, Jung *et al.*, 2019 and Wee *et al.*, 2019 also provide extensive lists of tools for long-read data.

Regardless of the sequencing technology, dealing with sequencing errors—identifying, correcting or even dismissing low-quality reads—before and/or during both genome and transcriptome assembly is of high importance (Bokulich *et al.*, 2013; Edgar and Flyvbjerg, 2015). If not treated appropriately, they can be detrimental. Some assemblers have been developed jointly with a sequencing technology which optimises treating peculiarities arising from the sequencing and, vice versa, which help optimise the sequencing experiment to obtain good input data for the assembler (Fryslie, no date; Butler *et al.*, 2008).

It is worth mentioning that a whole genome or transcriptome assembly process, no matter the approach chosen, requires a non-negligible amount of computational resources. Assemblers typically heavily exploit efficient data structures and rely on various optimisation algorithms to reduce memory

consumption and break the big problem into smaller problems which can be computed in parallel (Ocaña and de Oliveira, 2015). Yet, the large amounts of input sequencing data and complexity of the problem require memory and time which often can be satisfied only on high-performance computing clusters.

1.1.3 Assembling challenges

As indicated in the previous section, building an assembly from a set of raw reads is challenging. A typical assembly contains fully and partially assembled sequences, with or without assembling errors within them. It is a consequence of biological complexities deriving the data, imperfect sequencing technologies and assembly heuristics attempting to deal with the data (Martin and Wang, 2011). In this section, we provide a brief overview of the most frequent *de novo* genome and transcriptome assembling challenges and the resulting artifacts. However, reference-based assembling is also prone to errors due to fragmentation and mistakes in the reference sequence(s) and due to true biological differences between the reference assembly and the one being assembled, thus leading to the same types of artifacts. If not detected and corrected, misassemblies lead to misannotations[5] and adversely affect downstream analyses (Kremer, McBride and Pinto, 2017).

The major challenge for genome assemblers lies in repetitive regions—highly similar or identical sequences present at different locations in the genome (Treangen and Salzberg, 2011; Meltz Steinberg *et al.*, 2017). Typically, repetitive regions make up the majority of eukaryotic genomes (Biscotti, Olmo and Heslop-Harrison, 2015), and in plants can take up even more than 80% of the genome—as indicated in maize and wheat analyses (Schnable *et al.*, 2009; International Wheat Genome Sequencing Consortium (IWGSC), 2014). Even bacterial genomes can consist of almost 40% repeats (Cho *et al.*, 2007).

[5] More about genome and transcriptome annotation in section 1.2

The problem of assembling repeats is particularly pronounced for reads coming from second-generation sequencers as their short length does not span the whole repetitive region (Treangen and Salzberg, 2011). Typical *de novo* misassemblies caused by repeats include fragmentation of repetitive regions, rearrangements (Fig. 1.1b), inversions (Fig. 1.1c), collapsing tandem repeats (Fig. 1.1e), collapsing interspersed repeats and repeat expansions (Fig. 1.1f) (Phillippy, Schatz and Pop, 2008; Treangen and Salzberg, 2011). Furthermore, repeat-flanking regions often get misassembled, too, or cannot be assembled into the same contig with the neighbouring repeat (Meltz Steinberg *et al.*, 2017). Collapsing repeats reduces both the size and complexity of a genome assembly while fragmentation and expansion increase. Orientation of mate pair reads and their insert size length can be beneficial for detecting and correcting misassemblies as illustrated in Figure 1.1a-c (Treangen and Salzberg, 2011; Li and Copley, 2013; Chaisson, Wilson and Eichler, 2015). In addition, read coverage analysis can indicate collapsing tandem repeats and repeat expansion (Fig. 1.1d-f). Longer third-generation reads can also help to obtain full-length repeats, especially if the repeat length is shorter than the read length (Acuña-Amador *et al.*, 2018; Sedlazeck *et al.*, 2018).

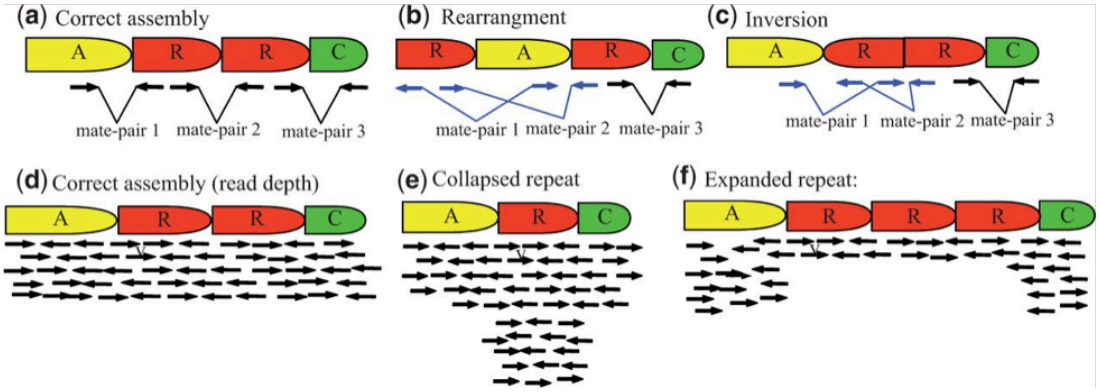


Figure 1.1: Examples of misassemblies caused by repetitive sequences (reproduced from: Muggli *et al.* (2015)[6]).

Boxes A and C represent non-repetitive regions and their orientations while R's depicts oriented identical repetitive regions. Arrows represent sequencing reads. Mate pair reads with correct orientation and insert length are depicted in black, otherwise in blue. a) Correct assembly (with correctly mapped mate pair reads). b) Rearrangement. c) Inversion. d) Correct assembly (with mapped paired-end reads depicting coverage). e) Collapsed repeat (with mapped paired-end reads depicting coverage). f) Expanded repeat (with mapped paired-end reads depicting coverage).

In genome sequencing, haplotypes of a diploid genome can be sequenced at the same time which poses difficulties in getting a single haploid-like assembly for highly polymorphic genomes[7,8] (Snyder *et al.*, 2015; Kyriakidou *et al.*, 2018). The problem is even more challenging for polyploid organisms which have more than two homologous copies of each chromosome, yet are still being represented by a single sequence (Meltz Steinberg *et al.*, 2017).

---

[6] Republished with permission of Oxford University Press (licence number 4841820489507), from Martin D. Muggli, Simon J. Puglisi, Roy Ronen and Christina Boucher, *Misassembly detection using paired-end sequence reads and optical mapping data*, Bioinformatics, 2015, Volume 31, Issue 12, Page i82.

[7] This common practice of single-sequence representation per set of homologous chromosomes stems from the high similarity of chromosome copies (Meltz Steinberg *et al.*, 2017).

[8] Some studies attempt to provide assemblies for each haplotype (e.g. Jones *et al.* (2004), Levy *et al.* (2007), Cao *et al.* (2015)). There are also tools for reconstruction of both haploid sequences from the input reads or assembly (e.g. Chin *et al.* (2016), Goltsman, Ho and Rokhsar (2017), Huang, Kang and Xu (2017), Koren *et al.* (2018)).

For example, The 1000 Genomes Project Consortium (2015) identified more than 88 million of putative single nucleotide polymorphisms (SNPs), indels and structural variants in the human genome, and Rimbert *et al.* (2018) identified 3.3 million putative SNPs in wheat.

Tools typically attempt to construct a mosaic of assembled sequences from homozygous regions and either consensus or randomly chosen sequences for heterozygous regions (Pryszcz and Gabaldón, 2016). However, assemblers usually collapse homozygous contigs[9] into a single contig, yet being cautious of *bona fide* duplications within a genome, they typically assemble the two heterozygous contigs separately and leave them both in the genome assembly (Fig. 1.2a) (Pryszcz and Gabaldón, 2016). Then, the assembler is unable to unambiguously merge contigs corresponding to neighbouring homo- and heterozygous regions. Similarly, a scaffolder might have difficulties in linking homo- and heterozygous contigs into scaffolds (Hunt *et al.*, 2014). Thus, the assembly remains fragmented with overestimated genome size. This is often referred to as allelic variation gaps (Chaisson, Wilson and Eichler, 2015). Consequently, shorter wrong gene models could get assigned to fragmented regions, synteny breaks, and putative paralogous genes and duplicated regions can get inferred.

[9] By homozygous contigs we mean contigs assembled from reads originating from homozygous genomic regions; analogous for heterozygous contigs.

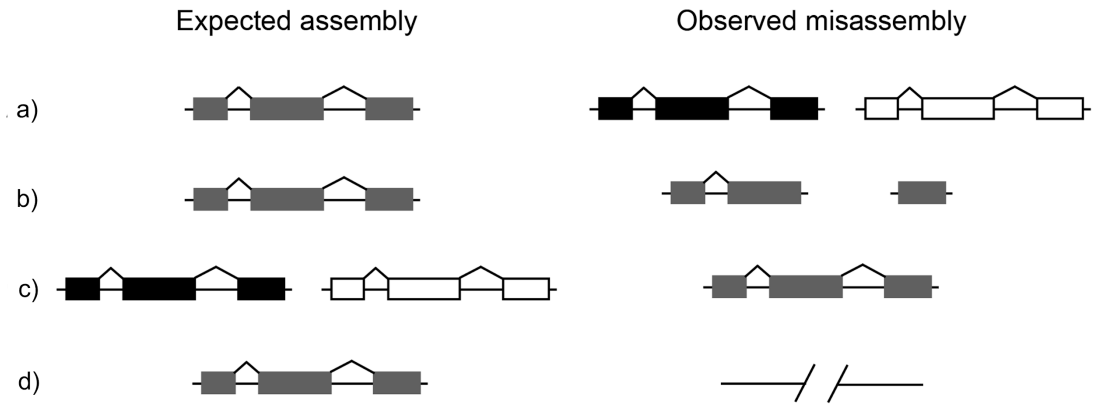


Figure 1.2: Various examples of misassemblies (reproduced from: Denton *et al.* (2014)[10]).

Boxes represent exons and straight lines between them introns. a) Erroneous sequence duplication. b) Sequence fragmentation. c) Collapsing sequences of paralogous genes. d) Missing gene sequence.

To identify some of these erroneous gene and region duplications, similarity searches for potential pairs of heterozygous regions, read coverage patterns and mate pair reads from the same individual, as well as sequencing parents and other individuals of the same species, can be employed (Kelley and Salzberg, 2010; Zhang and Backström, 2014; Pryszcz and Gabaldón, 2016). If available, synteny information can help scaffolding (Ghurye and Pop, 2019). Assemblies built from long reads are promising but can still contain sequencing errors interpreted as biological differences (Huang, Kang and Xu, 2017; Sedlazeck *et al.*, 2018; Rice and Green, 2019). Various software tools try to utilise these concepts in an automatic fashion. While some assemblers are specifically designed for heterozygous genomes (Chin *et al.*, 2016; Edge, Bafna and Bansal, 2017; Kajitani *et al.*, 2019), some genome assemblers implement a feature to deal with heterozygosity (Gnerre *et al.*, 2011; Safonova, Bankevich and Pevzner, 2014). There are also tools which attempt to reduce the redundancy in the already existing assemblies (Huang

[10] Used under the terms of Creative Commons Attribution International License 4.0 (https://creativecommons.org/licenses/by/4.0). Capital letters labelling the panels in the original figure were converted into lower case.

et al., 2012; Pryszcz and Gabaldón, 2016). However, none of the tools fully resolve the issues arising from heterozygosity.

Gene duplication can cause construction of unrealistic chimeric sequences, collapsing multiple sequences into one and sequence fragmentation (Denton *et al.*, 2014; Chaisson, Wilson and Eichler, 2015; Indrischek *et al.*, 2016).

A frequent artifact is the creation of chimeric sequences where reads derived from paralogous genes are merged together (Vukašinović *et al.*, 2014; Indrischek *et al.*, 2016). Consequently, annotation pipelines assign them unrealistic gene models. Such mis-joins in the assembly could be detected by analysing alignments of the input reads to the assembly. More precisely, multiple reads sharing a breakpoint in the alignment indicate a potential misassembly (Phillippy, Schatz and Pop, 2008).

Another common scenario is collapsing highly similar sequences derived from paralogous genes (Fig. 1.2c) which reduces the size and complexity of a putative genome (Denton *et al.*, 2014). Such sequences could be detected by examining coverage patterns of aligned reads.

Finally, sequences coming from paralogous genes might not get fully assembled due to assembler's inability to unambiguously extend contigs (Alkan, Sajjadian and Eichler, 2011; Chaisson, Wilson and Eichler, 2015). Gene sequences remain fragmented over multiple contigs which can end up on the same or different scaffolds (Fig. 1.2b) (Alkan, Sajjadian and Eichler, 2011). The subsequent annotation pipeline might then annotate each fragment as a separate gene model, thereby increasing the overall number of putative genes (Denton *et al.*, 2014; Salzberg, 2019). If fragments are on contigs which are too short (typically < 100 bp or < 200 bp), the contigs might get filtered out from the assembly (e.g. Zerbino and Birney (2008), Simpson *et al.* (2009), Simpson and Durbin (2012), Jackman *et al.* (2017)). Thus, the gene sequence might be partially or completely missing from the assembly and annotation (Fig. 1.2d).

Paired-end reads can help resolve some cases of fragmentation and extend contigs (Boetzer *et al.*, 2011; Renaut *et al.*, 2018). Putative full-length paralogous sequences can be reconstructed from long reads as well (Pollard *et al.*, 2018; Vollger *et al.*, 2019). Fragments of the same putative paralog may also lie on two interleaved scaffolds with non-overlapping contigs. Some annotation tools can recognise and correctly annotate such cases (Keller *et al.*, 2008; Indrischek *et al.*, 2016).

Other examples of genome misassemblies include indels (insertions and deletions) and base errors—due to assembler's heuristics or sequencing errors (especially for third-generation sequencers). Again, inspecting alignment patterns of input reads to the assembled contigs might be of help (Gurevich *et al.*, 2013; Hunt *et al.*, 2013; Zhu *et al.*, 2015; Kremer, McBride and Pinto, 2017; Rice and Green, 2019).

Some genomic regions might be challenging to sequence and assemble due to base pair composition (Baker, 2012; Benjamini and Speed, 2012; Ghurye and Pop, 2019). For example, AT- and GC-rich regions with reads affected by GC-content bias (due to PCR amplification in the sequencing protocol[11]) might lead to fragmented sequences, translocation errors and artificial tandem repeats if the bias is not reduced prior to assembling (Chen *et al.*, 2013). More complete and accurate assembly can be achieved with higher sequencing depth, yet high coverage also poses the risk of more assembly errors (Chen *et al.*, 2013).

Moving from contigs to a single gapless DNA sequence representation of a chromosome[12] is quite challenging. An assembler typically stops extending contigs due to an inability to choose among multiple possible options, usually stemming from the repetitiveness of the region (Hunt *et al.*, 2014; Ghurye and Pop, 2019; Rice and Green, 2019). Thus, the linking data used for

[11] There are also amplification-free sequencing protocols (Kozarewa *et al.*, 2009) and technologies, e.g. Helicos and Pacific Biosciences (Sam *et al.*, 2011).
[12] Depending on the study, it can be chromosome arm or, more generally, sequenced region.

scaffolding should be long enough to span the whole repeat (or difficult region) but short enough so as not to span multiple repeated sequences (difficult regions) (Ghurye and Pop, 2019). It should also allow for correct estimation of the distances between contigs. Usually additional data from multiple technologies is used for scaffolding. However, a scaffolder still might not be able to unambiguously and correctly place all contigs within a genome given the available evidence. Consequently, that affects gap filling, i.e. the reconstruction of sequences in the gaps between contigs, and the fragmentation remains. Some scaffolders do not even attempt to order and orient repeats (Koren, Treangen and Pop, 2011). The presence of heterozygosity in non-haploid organisms also complicates the goal of getting a single sequence assembly, as mentioned earlier.

Evaluations of *de novo* genome assemblies indicate that a large number of gaps is found in regions affected by GC-content bias, repeats and allelic variation (Vezzi, Narzisi and Mishra, 2012; Denton *et al.*, 2014; Meltz Steinberg *et al.*, 2017). Common types of gaps include: i) sequence coverage gaps or depth of coverage gaps, ii) segmental duplication-associated gaps, iii) satellite-associated gaps, iv) muted gaps, and v) allelic variation gaps mentioned earlier (Chaisson, Wilson and Eichler, 2015).

Sequence coverage gaps happen when no reads are sampled from a region while depth of coverage gaps occur when the reads are sampled with low coverage due to sequencing challenges (e.g. sequencing AT- and GC-rich regions with NGS technologies) (Chaisson, Wilson and Eichler, 2015; Meltz Steinberg *et al.*, 2017). It could be resolved by resequencing the problematic region (Meltz Steinberg *et al.*, 2017).

Segmental duplication-associated gaps are caused by the assembler's inability to unambiguously resolve possible scenarios caused by highly similar sequences (>90% sequence identity) (Chaisson, Wilson and Eichler, 2015). Hence, the assembler stops extending contigs and the region is represented by fragments flanked with gaps. The problem becomes even more challenging for structurally polymorphic segmental duplications in

diploid and polyploid organisms, increasing the probability of mistakenly merging reads and contigs derived from different alleles. Assembled sequences could be further extended with the help of additional long reads (Chaisson, Wilson and Eichler, 2015).

Satellite-associated gaps are associated with short tandem repeats (microsatellites), variable number of tandem repeats (macrosatellites) and centromeric satellite repeats (Chaisson, Wilson and Eichler, 2015). Based on reads‘ overlap, i.e. “piling up” of reads, the assembler is not able to keep extending a contig. Read coverage analysis can be helpful in determining the copy number of repeats but sequencing reads longer than the repetitive region is essential for accurate assembly. Centromeric, acrocentric and telomeric chromosome areas are abundant with these repeats, and being so challenging to reconstruct, the areas are typically not part of the assembly (Chaisson, Wilson and Eichler, 2015).

Muted gaps refer to ungapped sequences with a collapsed or truncated part typically present in many individuals (Chaisson, Wilson and Eichler, 2015). Beside assembly limitations in polymorphic and repetitive regions, muted gaps could arise from sequencing protocol. In particular, in clone-based sequencing, some DNA regions get deleted during cloning if they are toxic to bacteria (Chaisson, Wilson and Eichler, 2015; Meltz Steinberg *et al.*, 2017). Long-read technologies have proven to be useful for gap closing regardless of whether it arises from sequencing or assembling challenges (Sedlazeck *et al.*, 2018).

Some genome assembly challenges are encountered in assembling transcriptomes as well. In transcriptome assemblies, repeated sequences are usually shorter and appear with lower copy-numbers, still posing analogous challenges but to a lesser degree (Lima *et al.*, 2017; Hölzer and Marz, 2019). Some mistakes (e.g. collapsed tandem repeats, repeat expansion) cannot be detected and corrected in the same way as in the case of a genome assembly because here read coverage is related to the expression level and not copy-number (Martin and Wang, 2011). Reads

derived from transcripts expressed by different gene copies are present. If contigs corresponding to homologous transcripts are collapsed into a consensus contig, it can be detected by mapping input reads to the assembled contig and measuring nucleotide supports (Fig. 1.3, Family collapse) (Smith-Unna *et al.*, 2016). Insertions will not be supported by the evidence from input reads (Fig. 1.3, Unsupported insertion) (Smith-Unna *et al.*, 2016). Similarly, input reads will not support deletions within the contig introduced by an assembler and, more generally, incomplete transcript sequences (Fig. 1.3, Incompleteness) (Smith-Unna *et al.*, 2016). Again, base errors are present due to uncorrected sequencing errors or have been introduced as a consequence of assembler's heuristics (Martin and Wang, 2011). Some of them can still be detected and corrected with the reads at hand. Chimera can be detected if the expression levels of corresponding transcripts differ (Fig. 1.3, Chimerism), otherwise it is hard to distinguish them from true trans-spliced transcripts (Martin and Wang, 2011). Rearrangements and inversions can be detected from the input paired-end reads or additional mate pair reads by inspecting the insert size length and orientation of the reads as in the case of genome assembly (Fig. 1.3, Local misassembly) (Smith-Unna *et al.*, 2016). Like in genome assemblies, long-read sequencing can be useful in resolving challenging regions (Sedlazeck *et al.*, 2018).

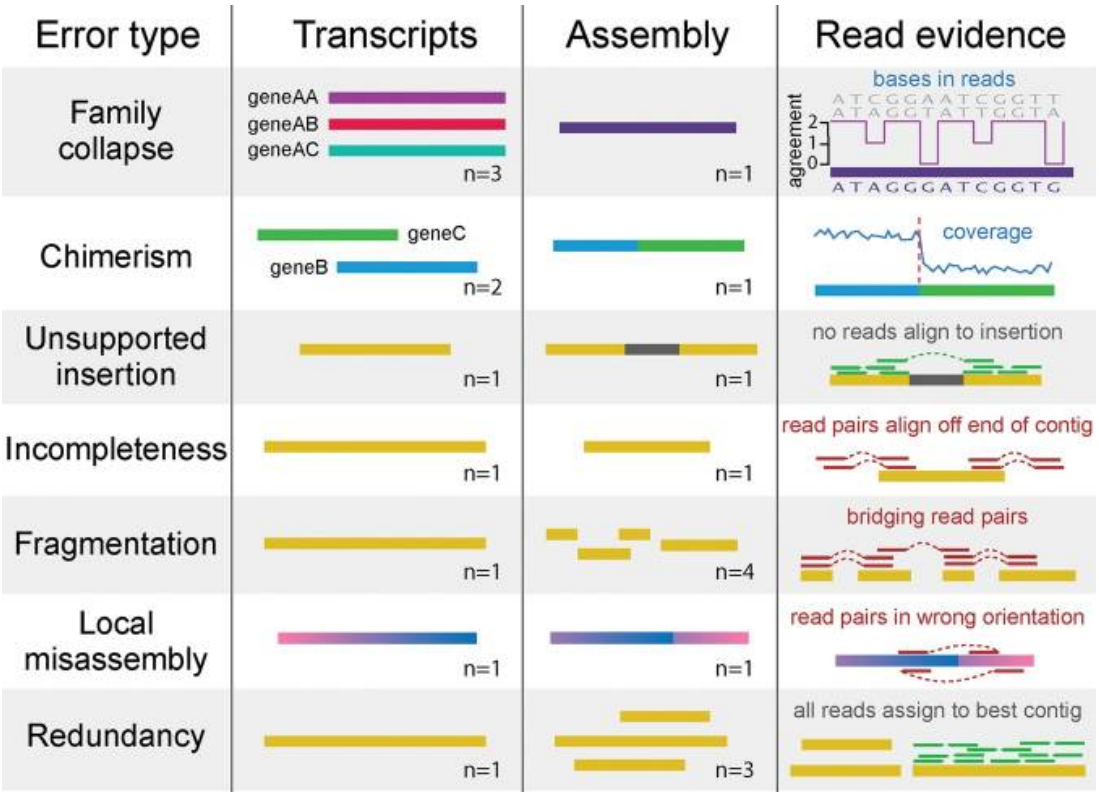


Figure 1.3: Examples of transcript misassemblies and their identification with paired-end reads used for assembly (reproduced from: Smith-Unna *et al.* (2016)[13]).

As already mentioned, genes are expressed at various levels which is reflected in the reads derived from corresponding RNA (Van Verk *et al.*, 2013). For example, some genes in plants can have more than five times higher expression level than others present in the same sample (Bräutigam *et al.*, 2011; Gowik *et al.*, 2011). Higher expression levels (which lead to higher sequencing coverage) are favourable for assembling more and longer transcript sequences (Zhao *et al.*, 2011) while sequences of poorly expressed transcripts (having lower sequencing coverage) are more likely to remain fragmented (Babarinde, Li and Hutchins, 2019). Furthermore, transcripts with low levels of expression might be considered as assembly artifacts and removed from the assembly (Hölzer and Marz, 2019). For fragments present in the assembly, fragmentation due to low expression can

[13] Used under the terms of Creative Commons Attribution International License 4.0 (https://creativecommons.org/licenses/by/4.0). No changes were made to the original figure.

be resolved with the help of read pairs mapping to two contigs (Fig. 1.3, Fragmentation) (Smith-Unna *et al.*, 2016). Researchers should also be aware of low coverage caused by sequencing GC-rich regions using protocols employing PCR amplification (Martin and Wang, 2011).

Particularly in eukaryotes, a gene can produce multiple transcript isoforms sharing exons (Fig. 1.4) and their reconstruction is one of the major transcriptome assembly challenges (Nilsen and Graveley, 2010; Steijger *et al.*, 2013). Transcript isoforms arise from alternative transcription start sites, alternative splice sites and alternative polyadenylation sites (Proudfoot, Furger and Dye, 2002). Some studies suggest that more than 90% of human multiexonic genes express transcript isoforms (Pan *et al.*, 2008; Wang *et al.*, 2008), and Reyes and Huber (2018) argue that the majority of them could be due to alternative transcription start sites and alternative polyadenylation sites. An assembled transcript isoform might contain extra or missing exons and include or exclude wrong unspliced introns. An assembler might co-assemble sequences of distinct transcript isoforms, as well as produce fragmented isoform sequences or miss an isoform altogether (Góngora-Castillo and Buell, 2013; Steijger *et al.*, 2013).

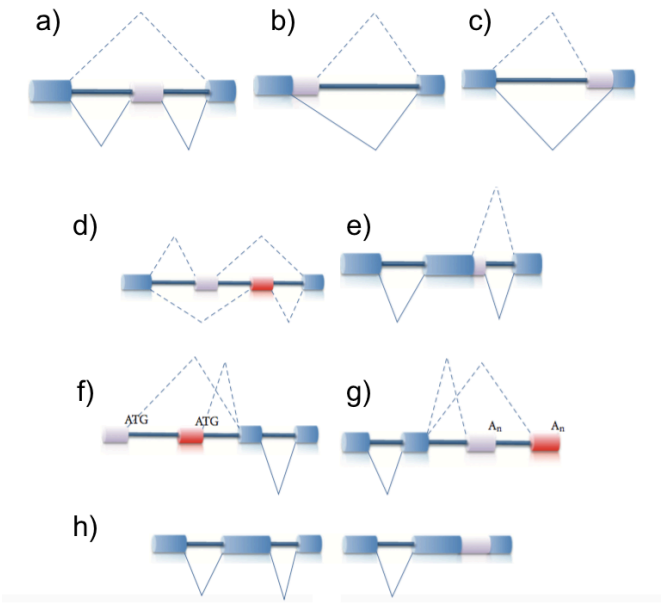


Figure 1.4: Transcript isoforms (reproduced from: Costa *et al.* (2010)[14]). Constitutive exons are depicted by blue boxes, alternatively spliced exons by red and violet boxes, and alternative splicing events by dashed lines. a) Canonical exon skipping. b) Alternative 5′ splicing. c) Alternative 3′ splicing. d) Mutually exclusive splicing. e) Intra-exonic splice site (exon depicted in the middle is partially excluded from the transcript). f) Alternative transcription start site (new alternative 5′ exon). g) Alternative polyadenylation site (new alternative 3′ exon). h) Intron retention.

As shown in Figure 1.5, read coverage (i.e. exon-level expression data) can help determining isoform sequences, yet due to heuristic solving of a complex combinatorial task, it might lead to missing putative isoforms. Information from paired reads and long-read sequencing help to obtain more complete putative isoform sequences and reveal more alternative putative isoforms of the same gene (Katz *et al.*, 2010; Sharon *et al.*, 2013; Bolisetty, Rajadinakaran and Graveley, 2015; Sedlazeck *et al.*, 2018). Additional information on isoform-specific exon-exon junctions and reads spanning exon-exon junctions can also help reconstructing transcript isoforms of the same gene (Góngora-Castillo and Buell, 2013).

[14] Used under the terms of Creative Commons Attribution International License 3.0 (https://creativecommons.org/licenses/by/3.0). Panel labelling was modified.

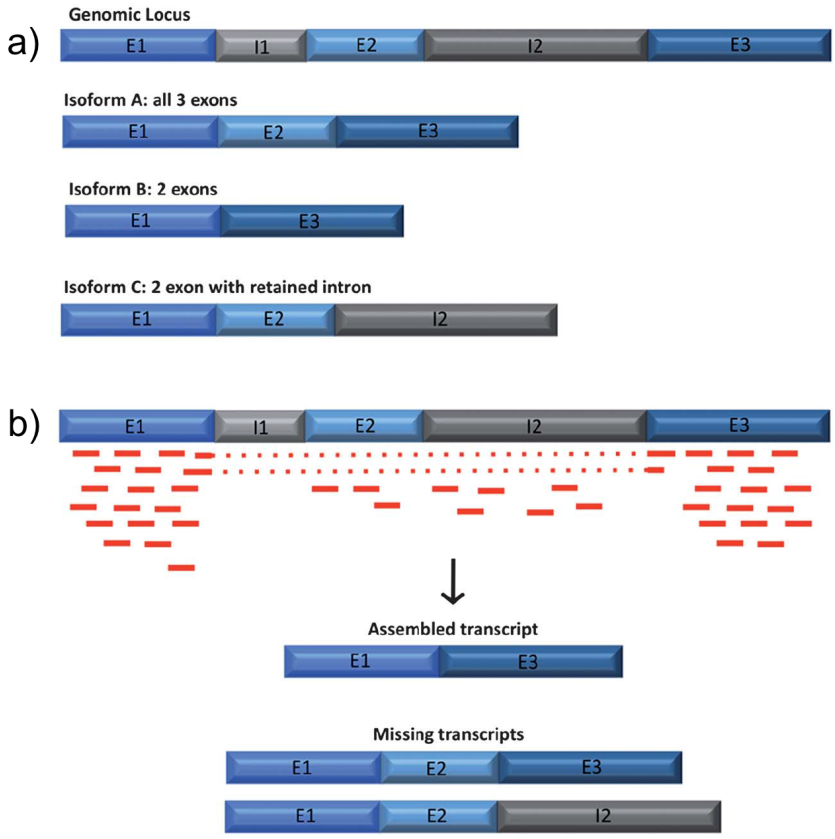


Figure 1.5: Missing transcript isoforms (reproduced from: Góngora-Castillo and Buell (2013)[15]).

Boxes E1, E2 and E3 depict exons while I1 and I2 depict introns. Red lines represent sequencing reads. a) A gene with three alternative isoforms. b) Read coverage misled to correctly assembling one transcript isoform sequence and missing two.

In general, low-abundance transcripts, especially if derived by complex splicing patterns, are more challenging to assemble and thus, more likely to be missing from the assembly or represented by fragments (Babarinde, Li and Hutchins, 2019).

As already mentioned in the previous section, the aim of transcriptome assembly is to reconstruct the sequences of all the expressed transcripts and

[15] Republished with permission of the Royal Society of Chemistry, from *Bioinformatics challenges in de novo transcriptome assembly using short read sequences in the absence of a reference genome sequence*, Elsa Góngora-Castillo and C. Robin Buell, volume 30, edition number 4, 2013; permission conveyed through Copyright Clearance Center, Inc. (licence ID 1039523-1).

their isoforms in a given sample (Martin and Wang, 2011). Thus, when dealing with non-haploid organisms, transcripts originating from both monoallelic and biallelic gene expression will be present in a sample, regardless of homo- or heterozygosity (Knight, 2004; Gimelbrant *et al.*, 2007; Filipczyk *et al.*, 2013; Eckersley-Maslin and Spector, 2014; Pinter *et al.*, 2015). High rates of heterozygosity pose difficulties for transcriptome assemblers as well (Voshall and Moriyama, 2018). In the case of biallelic gene expression, transcript sequences derived from heterozygous alleles can be fragmented over multiple contigs (Fig. 1.3, Fragmentation) while sequences from homozygous alleles tend to be collapsed due to their high similarity (Fig. 1.3, Family collapse) (Ruttink *et al.*, 2013; Farrell *et al.*, 2014; Ojeda *et al.*, 2019). Again, longer reads from third-generation sequencing technologies can span whole transcript length, and thus reduce fragmentation and increase assembly completeness, assuming corrected sequencing errors (Ruttink *et al.*, 2013; Ojeda *et al.*, 2019).

Some downstream applications, such as single-nucleotide polymorphism (SNP) discovery, require a reference transcriptome assembly such that each locus is represented by a single consensus allelic variant (Ruttink *et al.*, 2013; Stočes *et al.*, 2016). However, it is quite challenging to differentiate between sequences derived from allelic variants of the same locus and sequences derived from paralogs at different loci (Dlugosch *et al.*, 2013). If available, a proteome of closely related species can help to increase the contiguity and reduce the allelic redundancy of assembled transcripts (Ruttink *et al.*, 2013; Koning-Boucoiran *et al.*, 2015; Stočes *et al.*, 2016).

Being overcautious about duplicated genes, transcript isoforms and polymorphisms on the data with various gene expression levels and sequencing errors, an assembler might represent a transcript by multiple overlapping contigs (Fig. 1.3, Redundancy) (Liu *et al.*, 2014). In that case, input reads map to multiple contigs but are all assigned to the best representative (e.g. the longest contig) (Smith-Unna *et al.*, 2016).

Analysis of sequencing data revealed that gene expression, especially in eukaryotes, is much more complex than previously thought. For example, there are indications that 75–85% of the human genome is transcribed, of which only up to 2% is protein-coding (Djebali *et al.*, 2012). Some types of non-coding RNA (ncRNA), like ribosomal RNA (rRNA), transfer RNA (tRNA), small nuclear RNA (snRNA) and small nucleolar RNA (snoRNA), have established roles (Jensen, Jacquier and Libri, 2013). There are also many ncRNA molecules, with currently unknown functions, which are often said to result from the process of pervasive transcription (Kapranov *et al.*, 2007; Jacquier, 2009; Jensen, Jacquier and Libri, 2013; Wade and Grainger, 2014). Their transcription typically starts at unexpected places, such as intragenic regions and (non-coding) antisense strand, their expression levels are usually very low and it is believed that the majority of them are likely "junk" (Berretta and Morillon, 2009; Lybecker, Bilusic and Raghavan, 2014; Palazzo and Lee, 2015; Neri *et al.*, 2017).

Assembling the landscape of various RNA molecules is further complicated by imperfections of sequencing protocols. Researchers are usually interested in assembling transcripts from protein-coding messenger RNA (mRNA) which typically makes 1-2% of the total cellular RNA (Conesa *et al.*, 2016). Yet, usually the majority of cellular RNA, even more than 90%, is comprised of rRNA (Kukurba and Montgomery, 2015; Conesa *et al.*, 2016). Thus, a sequencing protocol typically includes either an mRNA enrichment or rRNA depletion step to improve the detection of mRNA transcripts (Kukurba and Montgomery, 2015). Due to its convenience and low cost, Poly(A)+ enrichment is preferable when applicable (Hrdlickova, Toloue and Tian, 2017). It can be used for RNA molecules containing poly(A) tail, such as most mRNAs and long non-coding RNAs (lncRNAs) in eukaryotes (Conesa *et al.*, 2016; Hrdlickova, Toloue and Tian, 2017). However, samples of low quantity or low quality and samples of mRNA molecules without poly(A) tail require rRNA depletion (Kukurba and Montgomery, 2015; Conesa *et al.*, 2016; Pereira, Imada and Guedes, 2017).

Unfortunately, these RNA sequencing protocols come with artifacts. For example, some studies indicate that current methods (often but not necessary including poly(A)+ selection) yield reads from both precursor mRNA (pre-mRNA) and mRNA (Zhang *et al.*, 2015). Assembling reads from pre-mRNA gives sequences of incompletely spliced isoforms (with retained introns)[16] which lead to artificial gene models in transcriptome annotation. Sequencing some of these spurious isoforms can be avoided using polyribosomal RNA-seq technology (Zhang *et al.*, 2015). An rRNA depletion protocol has been reported to lead to false models of alternatively spliced transcripts due to uneven sequencing coverage and skipped (i.e. unsequenced) exons (Sun *et al.*, 2013).

Importantly, despite poly(A)+ enrichment or rRNA depletion, some rRNA transcripts still get sequenced. Some of the corresponding reads can be detected by mapping the sequencing data to the databases of rRNA reads such as SILVA (Glöckner *et al.*, 2017) or 5SrRNAdb of 5S rRNA sequences (Szymanski *et al.*, 2016). There are also protocols for the enrichment of small ncRNAs, such as microRNA (miRNA), small interfering RNA (siRNA) and piwi-interacting RNA (piRNA) (Kukurba and Montgomery, 2015; Pereira, Imada and Guedes, 2017).

Unlike DNA, RNA sequencing can be strand-specific which allows for better sequence reconstruction of antisense transcripts in comparison to non-strand-specific sequencing protocols (Levin *et al.*, 2010; Tsai *et al.*, 2015).

So far, we have discussed assembling challenges for second-generation sequencing reads. Long third-generation sequencing reads have been mentioned at multiple places as a potential remedy. However, the sequencing and computational methods for third-generation sequencing data are still limited in terms of producing gapless genome assemblies and

[16] Not all splice isoforms with introns are products of pre-mRNA. However, Zhang *et al.* (2015) speculate that retained introns from pre-mRNA could make up a large portion of introns in those RNA-seq datasets containing reads from pre-mRNA and mRNA.

complete transcriptome assemblies. For example, Koren *et al.* (2017) reported complete genome assemblies for a small genome of *Escherichia coli* (~4.65 Mbp). Getting and making use of the long reads is an emerging field and beyond the scope of this thesis. Thus, we only briefly cover challenges encountered in their assemblies.

As indicated throughout this section, long sequencing reads can reduce assembling complexities arising from biological phenomena. However, these technologies suffer from a higher frequency of sequencing errors which can lead to erroneous and fragmented assemblies (Sedlazeck *et al.*, 2018; Jung *et al.*, 2019). Thanks to the existing error correction methods, e.g. PacBio reads derived from non-repetitive regions with ~20% sequencing errors can yield contigs with estimated accuracy of ≥99.9% but further improvements are still needed (Sedlazeck *et al.*, 2018). Both Pacific Biosciences and Oxford Nanopore Technologies have difficulties in sequencing homopolymeric and other low-complexity regions (Sedlazeck *et al.*, 2018; Jung *et al.*, 2019). Sequencing protocols also require further investigation as they perhaps prevent full-length sequencing of some transcripts or their isoforms (Tardaguila *et al.*, 2018; Wang *et al.*, 2019).

The advantage of the long reads is that they can cover longer repetitive regions, whole genes, transcripts, transcript isoforms or allelic variants (Mardis, 2017; Jin Lee and Pyo Hong, 2019). Longer reads also lead to longer contigs (Sedlazeck *et al.*, 2018). Yet, even assemblers for long reads typically stop extending contigs due to repeats and ambiguities arising from heterozygosity, especially for polyploid species (Sedlazeck *et al.*, 2018). Recent post-assembly tools attempt to reconstruct repetitive sequences in putative genomes based on discriminative statistical features among repeats (Bongartz, 2019) or based on high similarity of aligned error-corrected reads (Tischler-Höhle, 2019). Similarity of aligned error-corrected reads can also be used to resolve haplotype phasing (Tischler-Höhle, 2019). A more costly approach for a diploid genome is to sequence parental genomes using cheaper second generation approaches and employ them as a guide in

partitioning the offspring's long reads into putative haplotypes prior to genome assembly (Koren *et al.*, 2018). Reconstruction of heterozygous sites is further challenged by sequencing errors, and some approaches include error modeling and correction in order to infer biological differences more reliably (Beretta *et al.*, 2018).

The success of scaffolding is directly affected by the assembled contigs (their length, remaining mistakes, regions they span) and biases in the data used for scaffolding (e.g. lower coverage at fragile sites in BioNano Genomics maps, erroneous inversions in Hi-C-based approaches, limitations of Illumina sequencing such as GC-content bias in Hi-C and 10X Genomics data) (Sedlazeck *et al.*, 2018). If gap filling is subsequently performed, it is important to be aware of the biological complexities and sequencing biases which potentially led to the observed fragmentation (Sedlazeck *et al.*, 2018; Jung *et al.*, 2019).

1.1.4 Evaluation of assemblies

Finally, it is easy to come up with a method for merging sequences but it is hard to design a pipeline that does it correctly (Salzberg and Yorke, 2005). Furthermore, it is very challenging to evaluate the accuracy and completeness of the assembly given that the true sequences are unknown. Various measures have been used but none of them are perfect (Earl *et al.*, 2011; Bradnam *et al.*, 2013; O'Neil and Emrich, 2013). Thus, when evaluating an assembly, multiple measures should be considered and then interpreted in the light of their pros and cons.

Widely used statistics include Nx and Lx—both depending only on the assembly and not on the true genome or transcriptome (Gurevich *et al.*, 2013; Bushmanova *et al.*, 2016). Usually N50 and L50 are reported—the length of the shortest sequence in the assembly (contig or scaffold) such that sequences of that size or longer cover 50% of the assembly, and the number of sequences of that size or longer, respectively (Meltz Steinberg *et al.*,

2017). NG50 might be more informative as it is defined as the length of the shortest sequence in the assembly such that sequences of that size or longer cover 50% of the estimated genome size (Meltz Steinberg *et al.*, 2017). Measures taking into account the dynamic nature of transcriptomes have also been developed. For transcriptome assemblies, e.g. ExN50 is calculated on the normalised expression data in the same way as N50 but considering only the top *x%* most highly expressed transcripts (Haas, 2015; Geniza and Jaiswal, 2017).

In general, the size of the assembly and the number of annotated genes are often compared to the corresponding estimates which are challenging to obtain. For example, for each assembly they host, Ensembl database (Cunningham *et al.*, 2019; Howe *et al.*, 2020) provides information such as the number of basepairs, golden path length[17], number of annotated coding and non-coding genes, number of annotated pseudogenes, number of annotated gene transcripts and number of annotated structural variants. These numbers can then be compared to the estimations from the literature.

To assess the completeness of the assembly, approaches like BUSCO (Simão *et al.*, 2015), CEGMA (Parra, Bradnam and Korf, 2007) and coreGF (Van Bel *et al.*, 2012) check if the evolutionary conserved putative gene sets are present among the annotated genes which gives only a partial picture of the assembled and annotated gene space. Comparison to assemblies of closely related species or comparison between genome and transcriptome assembly can be powerful yet insufficient as they might not be sensitive enough to differentiate assembly errors from true biological differences or detect short-range rearrangements (Bradnam *et al.*, 2013).

The quality of the resulting assembly will depend on many parameters beside the biological complexity of the target species: from sequencing technology, experimental settings, read length, sequencing depth and other available data that can be used to improve assembly to the choice of

[17] Genome assembly size including estimated size of the gaps and excluding alternative sequences and pseudoautosomal regions.

assembly algorithms and other software tools, available computational resources, expertise on the species in question and the extent of manual curation. Depending on the study and available resources, some of the parameters can be optimised. Generally speaking, obtaining a high-quality assembly, especially for a non-model species, is still a herculean task, and the majority of sequencing projects reach the end by providing a “draft” genome and/or transcriptome assembly (Schliesky *et al.*, 2012; Denton *et al.*, 2014; Richards, 2018; Voshall and Moriyama, 2018).

1.2 Genome and transcriptome annotation

To gain new scientific knowledge from a genome or transcriptome assembly, the assembly typically needs to undergo annotation. There are two types of annotation: structural and functional annotation. Structural annotation encompasses identification of putative genes, gene products, their exons, introns in case of a genome assembly or retained introns in case of a transcriptome assembly, untranslated regions and regulatory elements, while functional annotation seeks to characterise the functional role (if any) of these various components within broader molecular, cellular, organismal processes (Yandell and Ence, 2012; Bolger, Arsova and Usadel, 2018; Dominguez Del Angel *et al.*, 2018). More specifically, the functional annotation comprises assignment of a wide range of biological information to the structurally annotated sequences such as name, domains, family, functional sites, biological functions, involvement in pathways and reactions, cellular location, etc., and usually exploits putative genes and gene products whose annotations have been obtained experimentally or *in silico* (Childs, 2014). Non-coding sequences can also be annotated, typically using different approaches (Bolger *et al.*, 2017).

Structural annotation starts with a computation phase (Yandell and Ence, 2012). The first step is the identification and masking of repeats (Yandell and Ence, 2012). The term “repeats” actually encompasses low-complexity sequences (e.g. homopolymeric runs of nucleotides) and transposable

elements (e.g. long interspersed nuclear elements (LINEs), short interspersed nuclear elements (SINEs)). They can be separately annotated with different tools (Smit, Hubley and Green, 2013-2015; Bergman and Quesneville, 2007). Then evidence from other sources is aligned to the assembly such as protein sequences, expressed sequence tags (ESTs) and RNA-seq data from the species being annotated and its closely related species (Brent, 2005). In the third step, tools for *ab initio* gene prediction are used (Jones, 2006). They typically rely on mathematical models and do not require external data as evidence. The models require training and parameter optimisation for the species under examination, which can be challenging. Sometimes a tool is trained on the sequences which will be then annotated (Lomsadze *et al.*, 2005; Ter-Hovhannisyan *et al.*, 2008). A tool can also come with a precomputed set of parameters (for example tuned on a publically available dataset of its authors' choice) but that might not be optimal for the current study because even the genomes of closely related species can have different properties such as intron lengths, codon usage and GC content (similar for transcriptomes) (Korf, 2004). However, if enough training data is available, *ab initio* tools can reportedly identify nearly all genes and less successfully intron-exon structures (~60-70%) (Yandell and Ence, 2012). Identification of boundaries can be improved, for example by using evidence from ESTs and long-read data (Allen, Pertea and Salzberg, 2004; Wei and Brent, 2006; Park *et al.*, 2017; Xia *et al.*, 2019). If additional information is used in the gene prediction process, the process is often called evidence-based or evidence-driven gene prediction (Liang *et al.*, 2009; Müller *et al.*, 2012; Yandell and Ence, 2012; Park *et al.*, 2017). Most of the gene prediction tools actually do not predict genes—they predict only the most likely coding sequence (CDS) for each gene and do not report untranslated regions (UTRs) (Childs, 2014). Furthermore, not all the tools report alternatively spliced variants in transcriptome assemblies (Childs, 2014). In both genome and transcriptome gene prediction, the tools might combine exons from multiple putative genes, skip exons or whole putative genes, as well as make false positive predictions (Childs, 2014).

The second phase of structural annotation is called annotation phase (Yandell and Ence, 2012). Evidence alignments and predictions from computation phase are manually or automatically combined to produce a final genome or transcriptome annotation with more accurate gene models (Zhang, 2002). In contrast to gene predictions, these outputs include UTRs. Transcriptome annotation outputs also include isoforms (Byrne *et al.*, 2019).

Structural annotations can be transferred to assemblies of the same or closely related species using reference-based annotation tools (see, e.g. Chatterji and Pachter (2006), Tcherepanov, Ehlers and Upton (2006), Otto *et al.* (2011))—a process referred to as similarity-based, reference-based or homology-based annotation (Gotoh, 2000; Wang, Chen and Li, 2004; Chatterji and Pachter, 2006). A reference-based annotation might not be limited to a single reference and it can be used in combination with other approaches to increase the confidence of annotations.

In practice, researchers usually run multiple gene prediction tools independently (computation phase) and then combine their outputs to choose annotation for each putative loci (annotation phase)—typically the most representative gene models for all obtained gene predictions given external evidence at hand. For example, the International Wheat Genome Sequencing Consortium (IWGSC) *et al.* (2018) annotated bread wheat genome assembly by combining the outputs of two independent computational pipelines (encompassing *ab initio*, evidence-driven and similarity-based gene model predictions) with supporting evidence (e.g. transcriptome data, putative homologs) and gene characteristics (such as length of coding region, proportion of canonical introns), and then integrating a manually curated set of gene models to obtain a more complete and less redundant annotation. Unver *et al.* (2017) produced a consensus set of gene models for the wild olive genome assembly from *ab initio* and homology-based approaches and supporting evidence from RNA-seq data.

Functional annotation is often based on sequence similarity between the sequence being annotated and annotated reference sequence(s). Existing

knowledge is propagated to the putative gene products under study based on inferred homology or orthology (Bolger, Arsova and Usadel, 2018; Dominguez Del Angel *et al.*, 2018). In cases with low sequence similarity, identification of protein motifs and domains can provide better functional annotation as they tend to be more conserved (Armstrong *et al.*, 2019). Such methods are typically based on representing already annotated putative gene families with multiple sequence alignments and hidden Markov models (HMMs) which are then used to annotate new sequences.

Non-sequenced data, both experimental and *in silico* predictions, can also be used for functional annotation or to increase the reliability of obtained annotations (Bolger, Arsova and Usadel, 2018). Expression data can help to determine if a putative protein is functional and provide insights into its function (Richardson and Watson, 2013; Childs, 2014). Methods employing "guilt by association" rule assign function based on co-expression with an annotated gene (Usadel *et al.*, 2009; Mutwil *et al.*, 2010; Tohge and Fernie, 2012; Obayashi *et al.*, 2018) or based on protein-protein interactions (Stelzl *et al.*, 2005; Sharan, Ulitsky and Shamir, 2007; Yu *et al.*, 2013).

Non-coding genome and transcriptome sequences can also be annotated. For example, repeats (low-complexity sequences and transposable elements) can be identified using *ab initio* methods and comparison to the already annotated sequences deposited in the databases (Saha *et al.*, 2008; Alexander *et al.*, 2010). Pseudogenes, thanks to their high sequence similarity to the putative functional parent gene, can be identified by searching against putative protein-coding sequences and inspecting additional properties such as premature stop codons, frameshifts and genomic context of the sequence to differentiate between putative genes and pseudogenes (Harrison, 2014). Databases of putative pseudogenes and evidence from literature can also be used (Chen, Ma and Zeng, 2011). Computational structural annotation of regulatory elements typically relies on identification of conserved sequence motifs—known or *ab initio* based on training data—with many false positive matches (Shlyueva, Stampfel and Stark, 2014). Due to the absence of coding constraints, the primary

sequence of non-coding RNA (ncRNA) tends to be less conserved and more heterogeneous (Yandell and Ence, 2012; Armstrong *et al.*, 2019). Furthermore, homology inference at the nucleotide level is more challenging than at a protein level. Yet, conserved secondary structure and motifs can help ncRNA structural annotation (Yandell and Ence, 2012). Functional annotation of non-coding sequences relies on experimental data (e.g. ChIP-chip, ChIP-seq and RNA-seq) (Alexander *et al.*, 2010). Co-expression with protein-coding genes across various experimental conditions can indicate involvement in the same biological processes or regulation of gene expression (Alexander *et al.*, 2010; Chen, Shi and Shi, 2017; Chekanova and Wang, 2019). Generally speaking, annotation of non-coding sequences is more difficult and less accurate than annotation of coding sequences (Yandell and Ence, 2012).

Having obtained a set of annotations, it is crucial to assess and curate them, as mistakes will affect downstream analyses and future annotations (Yandell and Ence, 2012; Childs, 2014). If an extensive high-quality reference annotation of the same species is available, it is possible to use it for assessment and correction of the newly obtained annotation. Approximations to precision and recall can be calculated at the level of gene models, annotated transcripts, exons and nucleotides, and annotation edit distance (AED) can be used to quantify changes between two annotations (Eilbeck *et al.*, 2009). However, usually a high-quality annotation of the same species is not available. Thus, the assessment depends on the complementary data from the same and/or different species—sequencing data, expression data, evidence from the literature. For example, a transcript from a reference species aligning to an annotated intron could indicate a misannotated exon (Childs, 2014). AED can also be used to quantify the agreement between an annotation and its supporting evidence (Eilbeck *et al.*, 2009). Measures such as percentage of exon-intron annotations supported by external evidence or percentage of annotations encoding known putative protein domains can also be calculated (Yandell and Ence, 2012). Similarly to assessing completeness of the assembly, BUSCO (Simão *et al.*, 2015) and CEGMA (Parra, Bradnam and Korf, 2007) can also be used to assess annotation.

Typically, evidence from a single or multiple sources is manually combined and judged, followed by manual correction of errors (Yandell and Ence, 2012; Childs, 2014).

Genome and transcriptome annotation is challenging. It depends on the quality of the assembly being annotated, reference annotation (its completeness, errors, evolutionary distance to the species under investigation), other available evidence (expression data, sequencing data, literature), tools being used and the extent of manual curation (Yandell and Ence, 2012; Richardson and Watson, 2013; Bolger, Arsova and Usadel, 2018). Assembly errors caused by uncorrected sequencing errors can yield structurally annotated sequences with artificial mutations or shorter sequences due to false stop codons. However, if the majority of the coding region is correctly assembled, most methods can deal with it (Bolger, Arsova and Usadel, 2018). Fragmented assemblies can induce fragmented and missing gene models (Bolger, Arsova and Usadel, 2018; Richards, 2018). Some annotation tools attempt to recognise and merge fragments derived from the same gene but spread across multiple contigs or scaffolds, yet many fragments remain misannotated as full-length gene models or unannotated due to their short length (Indrischek *et al.*, 2016). This directly affects the overall number of gene models in a putative genome. Similarly, biological phenomena such as gene fusion and fission introduce more challenges in the annotation. Nevertheless, the domains of those sequences (resulting from fragmentation, fusion or fission) can be correctly identified (Richardson and Watson, 2013). More reliable annotation is expected on assemblies derived from long third-generation reads, particularly if such a gene (or transcript) sequence is shorter than a read length (Wang *et al.*, 2019). These can also be employed as additional evidence for annotation of sequences assembled from shorter reads. Another challenging biological phenomena are alternative splicing, alternative transcription start sites and alternative polyadenylation sites yielding multiple transcript isoforms expressed by a single gene and present in a transcriptome assembly. Their annotation heavily depends on the quality of their assembly—partially and fully assembled isoform sequences, some of which contain mistakes as

substantial as extra or missing exons and retained introns, are subjected to annotation while some isoform sequences are completely missing. Again, long sequencing reads and corresponding assemblies can help avoiding structural annotation mistakes thanks to the improved identification of full-length isoforms and exon-intron boundaries (Abdel-Ghany *et al.*, 2016; Wang *et al.*, 2016, 2019). Non-coding sequences pose further annotation problems. Tools can predict coding sequences in non-coding regions, and methods have troubles with differentiating among ncRNA genes, spurious transcription and protein-coding genes with lower conservation rates (Yandell and Ence, 2012; Armstrong *et al.*, 2019). Some sequences can correctly be classified as coding or non-coding using measures of protein-scoring capacity from RNA-seq or mass spectrometry data (Chen, Shi and Shi, 2017). Some mistakes in structural annotation are discovered during functional annotation while some can cause further errors (Richardson and Watson, 2013; Childs, 2014; Dominguez Del Angel *et al.*, 2018). Furthermore, merging or updating annotations is not a trivial task—beside technical obstacles, such as different annotation formats, nomenclature and vocabularies, the main challenge is in measuring the quality of annotations, i.e. how close to the unknown truth they are (Wain *et al.*, 2002; Yandell and Ence, 2012).

Annotations, both structural and functional, should not be taken at face value but rather as hypotheses (Bolger, Arsova and Usadel, 2018). It is important to be aware that the absence of evidence does not mean that an annotated sequence is an artifact (in case of structural annotation) or does not carry a certain function (in case of functional annotation). If no annotated sequence carries the function of interest, it could be that such a sequence is missing from a dataset due to assembling errors such as fragmentation or collapsing highly similar sequences into one. Similarly, the presence of annotation does not guarantee its biological existence or its ascribed functional annotation (e.g. artificial chimeric sequences due to assembler's heuristics). Existing structural annotations can benefit from third generation sequencing data which is becoming available for more and more species (Cook *et al.*, 2019). Experimental data could confirm functional annotation, yet that requires more

resources and the majority of annotations have not been experimentally verified (Alexander *et al.*, 2010; Yandell and Ence, 2012; Richardson and Watson, 2013; Bolger, Arsova and Usadel, 2018).

More details on genome and transcriptome annotation including extensive lists of data and software resources can be found in the publications cited throughout this section, as well as for example in Sleator (2010), Garber *et al.* (2011) and Hosmani *et al.* (2019).

1.3 Plant assemblies

Plants are staple food for herbivores and omnivores. With increasing demands for food production and adjustment to climate changes, they are of high interest to the scientific community (Bevan *et al.*, 2017). Plant sequences are crucial for obtaining theoretical knowledge on evolutionary relationships and protein function. This knowledge can then be translated into agricultural and biotechnological innovations, such as accelerating the quest for traits, improving stress resistance, seeds and yield, and developing more efficient pesticides (Scheben, Yuan and Edwards, 2016; Bolger *et al.*, 2017; Hu, Scheben and Edwards, 2018). The value and potential benefits of high-quality genome and transcriptome plant data cannot be overestimated, yet the challenges of obtaining it cannot be underestimated.

1.3.1 Current state of plant assemblies

The challenge of genome assembly is particularly acute in plants. In their 2012 review, Claros *et al.* report that only around 80 000 plant species out of more than 370 000 known plants have at least one sequence in GenBank (Benson *et al.*, 2012), with *Arabidopsis thaliana* and *Oryza sativa* being of high quality and the rest being drafts. As of December 2019, at least 67 annotated plant genomes have been deposited in Ensembl Plants database (Howe *et al.*, 2020), 456 in NCBI (Sayers *et al.*, 2019) and 362 in plaBiPD

database (Usadel lab—Jülich Research Centre/RWTH Aachen University, 2014-2019)—only a minority of them being of good quality. The key issue is that plant genomes tend to be large, complex and heavily redundant (Carmona *et al.*, 2015; Jiao and Schneeberger, 2017). Data from such genomes frequently result in fragmentary assemblies with fragments derived from a single gene dispersed over two or more contigs. Consequently, the fragments get annotated as separate shorter gene models and the gene counts are overestimated (Schliesky *et al.*, 2012; Denton *et al.*, 2014). Fragmentary gene models do not only lack sequence information—they have been shown to cause problems in downstream analyses, such as in tree inference (Sayyari, Whitfield and Mirarab, 2017) and orthology inference (Dalquen *et al.*, 2013; Train *et al.*, 2017), limiting biological and biotechnical research and innovation.

A big genome size is not a problem in itself; it is the complexity of the genome that poses problems, as explained throughout section 1.1.3. A lot of difficulties are caused by repetitive sequences which can make up to 90% of a genome, the majority of them being transposable elements (Mehrotra and Goyal, 2014). Gene duplication poses difficulties to assemblers which have to distinguish reads coming from different genes. Some studies suggest that 57-70% of the extant flowering plants and up to 80% of all extant plants could be recent polyploids[18,19] (Otto, 2007; You *et al.*, 2018) which adds another layer of complexity to their genome and transcriptome assembling. Recently formed polyploid organisms have three or more sets of homologous chromosomes derived from the same (autopolyploid) or different (allopolyploid) species (Glover, Redestig and Dessimoz, 2016; Kyriakidou *et al.*, 2018; Voshall and Moriyama, 2019). In autopolyploid organisms, sets of

[18] Paleopolyploid events—polyploid events that took place more than a few millions years ago (Pfeil *et al.*, 2005)—are not expected to cause assembly troubles as the majority of duplicated genes undergo gene silencing within a few million years followed by gene loss (Lynch and Conery, 2000; Adams and Wendel, 2005). A good example is *Arabidopsis thaliana* (Kaul *et al.*, 2000). A study on soybean genome indicates no major troubles for whole genome assembly caused by paleopolyploidy (Schlueter *et al.*, 2007).

[19] Polyploidy has also been inferred in insects, fish, amphibia, reptiles and even red and golden viscacha rats (Otto, 2007).

homologous chromosomes are the consequence of whole genome duplication, whereas in allopolyploids, hybridisation between two species was followed by genome duplication—quite challenging situations for assemblers. Furthermore, studies suggest that polyploidisation can increase levels of heterozygosity (Kyriakidou *et al.*, 2018). In the attempt to avoid collapsing two similar sequence copies into one, genome assemblers might fail to recognise heterozygosity when all haplotypes of a diploid or polyploid genome are sequenced (Kelley and Salzberg, 2010; Zhang and Backström, 2014). The same applies for transcriptome assemblies if they aim for a single consensus transcript sequence per locus (Ruttink *et al.*, 2013; Stočes *et al.*, 2016). In transcriptome assemblies, assemblers have to distinguish reads coming from different transcript isoforms (Martin and Wang, 2011). Consequently, transcriptome assemblers have to detangle sequences of isoforms arising from paralogous genes, as well. The quality of the assemblies directly affects the quality of its annotation, as discussed in section 1.2. For example, for fragments derived from the same gene, protein domains can be correctly annotated yet the fragments typically remain annotated as separate gene products. On top of all this, there are also various low complexity repetitive sequences such as rDNA units, satellites, microsatellites, telomeric sequences contributing to the already highly intricate annotation problem (Claros *et al.*, 2012). Even single-copy regions (flanked by repeated sequences) can be tricky to annotate as there is a lot of variation in their length (Claros *et al.*, 2012). Correctly assembled sequences of expressed pseudogenes are difficult to distinguish from their parent gene, and thus might be annotated as a gene. Some of the assembling and annotation difficulties can successfully be resolved by employing paired-end reads, mate pair reads with long enough insert size, long reads, expression data and other experimental evidence, at the cost of more resources. Thus, many putative genomes remain in draft state (Mukherjee *et al.*, 2017).

Beside biological, there are also technical difficulties hindering plant sequencing and assembling (Claros *et al.*, 2012). The first challenge is to extract enough high-quality genetic material from the plant which is crucial for sequencing library preparation. During the sequencing experiment,

sequences can get contaminated by the wet-lab manipulations. The resulting samples can also contain data from organisms living on the plant or the human performing the experiment. Depending on its origin and (short) read length, contamination might be very challenging to identify. If there is a reference assembly of the species being assembled, sequencing reads that cannot correctly map to it could be derived from contaminants (Sangiovanni *et al.*, 2019). In the absence of a reference assembly, reads can be mapped against databases containing reads, gene or transcript sequences derived from potential sources of contamination (Schmieder and Edwards, 2011). Analogously, indications for contamination can be detected once the assembly is obtained (Alkan, Sajjadian and Eichler, 2011)[20]. Finally, as already mentioned earlier, second- and third-generation sequencing technologies are error prone (reported ~0.1-15%, 11-38%, respectively) which is something assemblers and assembly pipelines have to take into account and not confuse with true biological differences. The capabilities and limitations of assemblers can play a key role when it comes to assemblies of novel lineage-specific genes versus misassemblies.

Given all biological and technical challenges of sequencing, assembling and annotation, and the limited resources, it is not surprising that the majority of sequencing projects reach their end providing only a draft of a putative genome. A draft assembly can already aid research projects concerned with sufficiently well assembled parts of the genome. Indeed, many researchers are interested only in the coding regions of a genome and published draft assemblies can satisfy their needs (Thomma *et al.*, 2016). However, researchers should be cautious of the fragmentation and its potential underlying reasons (described in section 1.1.3). Furthermore, non-coding regions of a genome tend to be quite challenging for assemblers and consequently, represented by fragments with gaps in between them, if present at all in the assembly (see section 1.1.3). Altogether, to allow for more accurate and convenient downstream analyses of both coding and

[20] Within-species contamination, e.g. human contamination of a human sample, is even more challenging to detect. Jun *et al.* (2012) propose analysing sequencing data and/or array-based genotype data.

non-coding regions, it is necessary to improve current sequencing and assembling methods.

1.3.2 Bread wheat

The staple food for 30% of humanity, rich in protein, carbohydrates and minerals, common bread wheat, *Triticum aestivum*, is one of the most important crop species accounting for more than 95% of wheat grown worldwide (Choulet *et al.*, 2014; Pfeifer *et al.*, 2014). With the ongoing increase in the human population and corresponding demands for better yields, adaptation to climate changes and increased demands for biofuels, scientists, breeders and growers are in need of high-quality reliable resources to enhance biotech innovation (International Wheat Genome Sequencing Consortium (IWGSC), 2014). A good genome assembly would aid researchers to understand the molecular basis of phenotypic variation and help them identify candidate genes associated with favourable traits. This could then lead to the development of new cultivars with increased yield and improved resistance to biotic and abiotic stresses, i.e. higher bread wheat production (International Wheat Genome Sequencing Consortium (IWGSC), 2014).

Bread wheat has a large, complex and redundant genome. Today's allohexaploid AABBDD genome (6x=2n=42) is believed to be the result of three hybridization events which took place after the lineages *Triticum* (A) and *Aegilops* (B) diverged from a common ancestor (~6.5 million years ago) (Fig. 1.6) (Marcussen *et al.*, 2014). The first hybridization event probably took place ~5.5 million years ago between the A and B genome lineages and led to the origin of the D genome lineage. The second hybridization event likely occurred sometime between 0.58 and 0.82 million years ago between *Triticum urartu* (AA) and an unknown close relative of *Aegilops speltoides* (BB) producing allotetraploid emmer wheat, *Triticum turgidum* (AABB). Finally, the third hybridization probably happened less than 0.4 million years ago between *Triticum turgidum* (AABB) and *Aegilops tauschii* (DD)

(Marcussen *et al.*, 2014) giving rise to today's AABBDD genome. The three putative homeologous subgenomes (A, B, D), each ~5.5 Gbp in length and comprised of seven pairs of chromosomes, construct the highly redundant, 17 Gbp long putative bread wheat genome[21] which is mainly (>80%) composed of highly repetitive transposable elements (International Wheat Genome Sequencing Consortium (IWGSC), 2014).

[21] In 2018, IWGSC estimated a mean genome size ~15.76 Gbp (International Wheat Genome Sequencing Consortium (IWGSC) *et al.*, 2018)

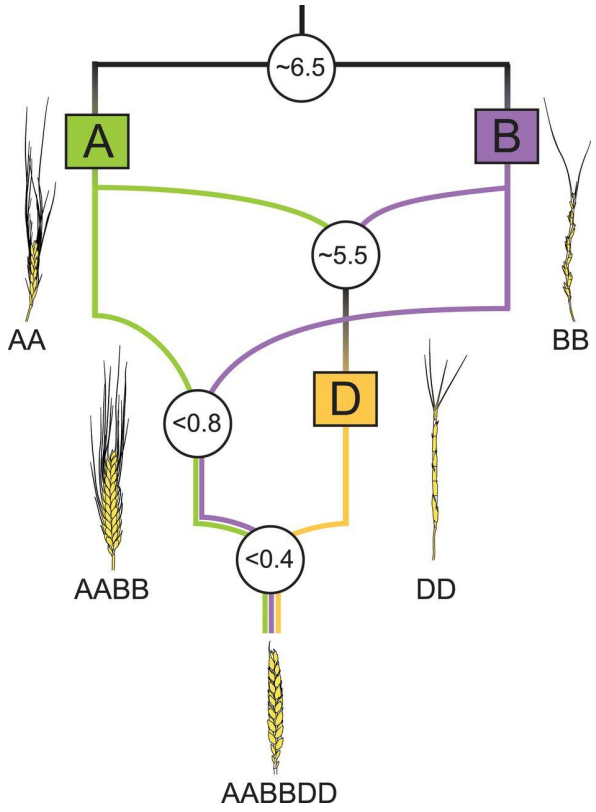


Figure 1.6: Evolutionary history of *Triticum aestivum* (AABBDD) (reproduced from: Marcussen et al. (2014)[22]).

Circles with numbers depict estimated times of: i) divergence of the A and B genome lineages from a common ancestor ~6.5 million years ago, ii) first hybridization event between the A and B genome lineages leading to the D genome lineage ~5.5 million years ago, iii) second hybridization event between *Triticum urartu* (AA) and a close relative of *Aegilops speltoides* (BB) leading to emmer wheat, *Triticum turgidum* (AABB), iv) third hybridization event between *Triticum turgidum* (AABB) and *Aegilops tauschii* (DD) leading to bread wheat, *Triticum aestivum* (AABBDD).

[22] From Thomas Marcussen, Simen R. Sandve, Lise Heier, Manuel Spannagl, Matthias Pfeifer, The International Wheat Genome Sequencing Consortium, Kjetill S. Jakobsen, Brande B. H. Wulff, Burkhard Steuernagel, Klaus F. X. Mayer and Odd-Arne Olsen, *Ancient hybridizations among the ancestral genomes of bread wheat*, Science, 345(6194), p. 1250092. Reprinted with permission from AAAS. Licence number 4841830781974.

For a long time, the size and complexity of the bread wheat genome caused doubts as to the possibility of sequencing and assembling it. The first breakthrough happened in 2008 when Paux *et al.* established a physical map of the largest chromosome, 3B, and sequenced it using BAC-by-BAC strategy (Choi and Wing, 2000; Lander *et al.*, 2001). The reads coming from cultivar Chinese Spring were assembled into 1,036 contigs of average length 783 kbp and N50 contig size of 602 kbp. They were anchored with 1,443 molecular markers. The assembly was believed to represent ~82% (811 Mbp) of the 995 Mbp long chromosome. Importantly, the study laid the methodological foundation for feasible sequencing and assembling of large and complex genomes.

The first bread wheat whole genome assembly was released in 2012 (Brenchley *et al.*, 2012). Chinese Spring reads were obtained by whole-genome shotgun sequencing (Messing, Crea and Seeburg, 1981) and assembled in two ways yielding a highly fragmented assembly. Their final orthologous group assembly covered 437,512,281 bp and contained 949,279 contigs of mean length 460.89 bp and N50 of 481 bp. The low-copy-number sequences assembly managed to cover 3,800,325,216 bp with 5,321,847 contigs of mean length 714.10 bp and N50 of 884 bp. With the help of transcriptome assembly (93,340,842 bp in total, 97,481 contigs, contig mean length 957.53 bp, N50 1,325 bp) they estimated the number of genes to be between 93,900 and 96,300.

In 2014, the International Wheat Genome Sequencing Consortium (IWGSC) released another highly fragmented chromosome-by-chromosome shotgun assembly for the cultivar Chinese Spring (International Wheat Genome Sequencing Consortium (IWGSC), 2014). The sequencing approach allowed sequencing an arm of a single chromosome copy at a time (except for chromosome 3B) which reduced complexity of *de novo* assembling Illumina reads and allowed obtaining haplotype-resolved assembly (Šafář *et al.*, 2010; Doležel *et al.*, 2012). The assembly (version IWGSP1) was estimated to represent ~61% of the genome sequence covering 10,237.9 Mbp (out of 16,938 Mbp) with 10,880,661 contigs longer than 200 bp whose N50 was

2,292 bp and average length 940.42 bp. Number of contigs varied across chromosome arms—from 88,542 on chromosome arm 6DS (N50 4,297 bp) to 508,239 on chromosome arm 2DL (N50 701 bp) and 546,922 on chromosome 3B (N50 2,655 bp). Chromosome arm 3DS had the smallest N50 of only 515 bp. The assembly allowed annotation of 124,201 genes with high confidence: 55,249 (44%) functional genes and 68,952 (56%) genes which seemed to be fragmented in the assembly. The latter ones were either structurally annotated or classified as pseudogenes and fragmented gene sequences.

The same year, Choulet *et al.* (2014) published a high-quality reference sequence of chromosome 3B, also for the Chinese Spring cultivar. An individual chromosome was flow-sorted and sequenced using BAC pooling strategy (Steuernagel *et al.*, 2009). The 2,808 scaffolds (N50 892 kbp) that were assembled represent 833 Mbp, i.e. ~94% of estimated 886 Mbp-long complete sequence and carry 5,326 (73%) annotated full genes and 1,938 (27%) likely pseudogenes or gene fragments.

Over subsequent years, assemblies with improved contiguity were released by Chapman *et al.* (2015), Clavijo *et al.* (2017) and Zimin *et al.* (2017).

Finally, in 2018, the International Wheat Genome Sequencing Consortium (IWGSC) released a fully annotated reference assembly for the bread wheat cultivar Chinese Spring (International Wheat Genome Sequencing Consortium (IWGSC) *et al.*, 2018). *De novo* assembly of Illumina pair-end and mate pair reads was complemented with additional data (Illumina paired-end, Hi-C, Ion Torrent and Roche-454 sequencing, genetic, physical, radiation hybrid and Bionano optical maps). The assembly (IWGSC RefSeq v1.0) covered 14,547.3 Mbp—around 92% of the newly estimated genome size (mean size 15,764.4 Mbp) (International Wheat Genome Sequencing Consortium (IWGSC) *et al.*, 2018, section 2.8 in Supplementary Materials). Contig N50 was 51.8 kb with L50 of 81,427 while scaffold N50 was 7.0Mb with L50 of 571. Contigs and scaffolds were further linked into superscaffolds (N50 22.8 Mbp, L50 166), with 97% of them being assigned and ordered

within chromosomes and covering 14.1 Gbp. Thus, ~90% of the genome was represented by superscaffolds, 76 superscaffolds per chromosome on average with the largest superscaffold spanning 166 Mbp. The annotation (IWGSC RefSeq v1.1) contained 107,891 highly-confident gene models, of which 90,919 (~84.3%) had a functional annotation as well.

1.3.3 Wild olive

The olive tree is an important fruit crop in the Mediterranean basin which holds 90% of all olive groves in the world yielding 90% of global olive oil production (Kole, 2011). Olive oil has been part of Mediterranean cuisine for thousands of years (Riley, 2002). Today it is consumed worldwide and appreciated for its health benefits (Estruch *et al.*, 2013, 2018). Beside nutrition, olive tree products are also used in pharmacy and cosmetics.

Cultivated olive tree, *Olea europaea* L. subsp. *europaea* var. *europaea*, is thought to have been domesticated from the wild olive, *Olea europaea* var. *sylvestris*, also called oleaster, and the domestication, diversification and selection of olives for cultivation is not well understood (Besnard, Terral and Cornille, 2018; Gros-Balthazard *et al.*, 2019). Genomic data could help understand the underlying processes and their consequences, as well as facilitate exploiting the pool of wild genes to improve disease and stress resistance of new cultivars (Kole, 2011).

Oleaster is a diploid species (2n=46) (Rugini *et al.*, 2011) and, at time of writing, its evolutionary history is still not well understood. A group of researchers who sequenced its only available putative genome to date found indications for two (oleaster) lineage-specific whole genome duplication events, ~28 million years ago and ~59 million years ago (Unver *et al.*, 2017). A more recent study (Julca *et al.*, 2018)[23] speaks of potentially three polyploidization events: i) allopolyploidization at the base of the Oleaceae

[23] They included the oleaster genome of Unver et al. (2017).

family between 33 and 72 million years ago, possibly with a non-Oleaceae Lamiales species, ii) allopolyploidization at the base of the Oleeae tribe between 14 and 33 million years ago, potentially with a relative of *Jasminum sambac*, and iii) olive-specific whole genome duplication (autopolyploidization) ~10 million years ago (Fig. 1.7).

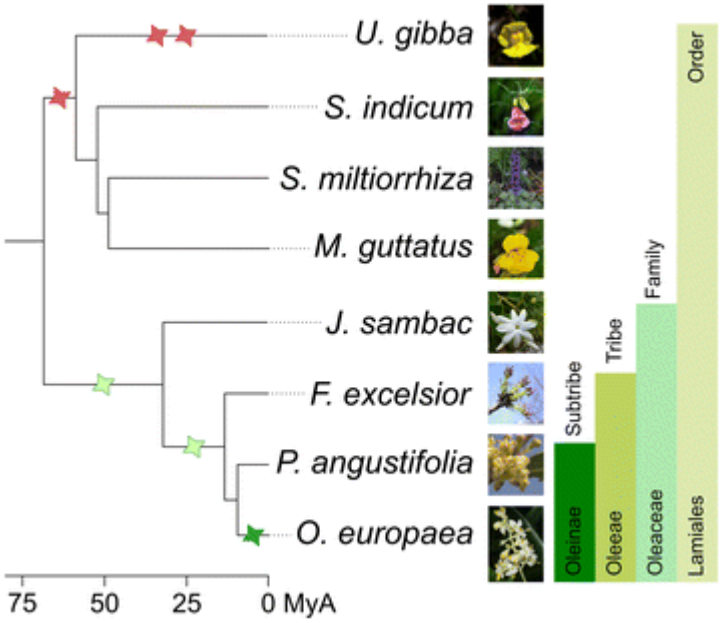


Figure 1.7: Newly estimated whole-genome duplications (green stars) and whole-genome duplications from the earlier literature (red stars) in Lamiales clade (reproduced from: Julca *et al.* (2018)[24]).

The putative genome sequence of *Olea europaea* var. *sylvestris* became available in 2017 (Unver *et al.*, 2017), thanks to the efforts of the International Olive (*Olea europaea*) Genome Consortium (IOGC). Illumina whole genome shotgun sequencing reads were assembled using SOAPdenovo (Li *et al.*, 2010) into 1.48 Mbp long assembly[25] consisting of 11,497 contigs with N50 of 25,485 bp. Contigs were organised into 1,448 scaffolds with N50 length 228,620 bp. The assembly allowed annotation of

[24] Used under the terms of Creative Commons Attribution International License 4.0 (https://creativecommons.org/licenses/by/4.0). No changes were made to the original figure.
[25] Authors estimated genome length to be 1.46 Mbp while Loureiro *et al.* (2007) estimated it to be 1.56 Mbp.

50,684 protein-coding genes. The study also reported that highly repetitive DNA, mainly transposable elements, made up 51% of the putative genome. Based on their data, the authors estimated heterozygosity rate of 1.3% (Unver *et al.*, 2017, Fig. S2)[26].

Other useful sequencing data for studying wild olive include genome assembly of domesticated olive tree *Olea europaea* L. subsp. *europaea* var. *europaea* cv. Farga (Cruz *et al.*, 2016), putative repetitive sequences from cultivar *Olea europaea* L. subsp. *europaea* var. *europaea* cv. Leccino (Barghini *et al.*, 2014), transcriptome data from cultivars Arbequina, Lechin de Sevilla, Picual and Picual x Arbequina cross (Munoz-Merida *et al.*, 2013) and *Olea europaea* L. reproductive transcriptome database ReprOlive (Carmona *et al.*, 2015), to name a few.

1.3.4 Cassava

Globally the most important root crop—cassava, i.e. *Manihot esculenta* Crantz—is a staple food for more than 700 million people (Sayre *et al.*, 2011). With a starch content between 20 and 40%, and favourable characteristics such as little input for big yield, high drought tolerance, adaptability to diverse environments and roots' capability to be left in the ground for months before harvesting, cassava demonstrates high potential for carbohydrate production and is a desirable source for bioenergy production (Zainuddin *et al.*, 2012). Unfortunately, it has low protein and micronutrient content. Furthermore, it is susceptible to bacterial and viral diseases which limits its current production (Bart and Taylor, 2017). To gain a better understanding of the cassava genes and accelerate development of desirable varieties, it is crucial to obtain and employ genetic resources such as high-quality reference genome and transcriptome assembly.

[26] heterozygosity rate = (#heterozygous loci / genome size) * 100 (personal correspondence with Turgay Unver)

Cultivated cassava, probably domesticated over 6000 years ago from the wild *Manihot esculenta* ssp. Flabellifolia (Wang *et al.*, 2014), is another example of a challenging assembly problem. Bredeson *et al.* (2016) found indications that its ancestor underwent a whole-genome duplication ~35-47 million years ago. Eighteen chromosomes form a 772 Mb long highly heterozygous diploid genome which contains a lot of repetitive regions, many of them transposable elements (Awoleye *et al.*, 1994; Wang *et al.*, 2014; Bredeson *et al.*, 2016). For example, assembly v6.0 with v6.1 annotation is 582.3 Mbp long, contains 299.3 Mbp of repetitive sequences, more than half being putative transposable elements, and the remaining unassembled ~200 Mbp are estimated to span mainly repeats and less than 1% of putative genes (Bredeson *et al.*, 2016). Not surprisingly, cassava is another example of long-lasting efforts to obtain a high-quality reference assembly.

Although the beginnings of efforts to sequence cassava date back to 2003, with the first pilot project being accepted in 2006, the first genome assembly and annotation were publically released in late 2009 (Prochnik *et al.*, 2012). Scientists from the University of Arizona, 454 Life Sciences and the US Department of Energy—Joint Genome Institute performed a combination of 454-based whole genome shotgun (Margulies *et al.*, 2005) and Sanger sequencing (Sanger, Nicklen and Coulson, 1977) to obtain reads from a 3rd-generation inbred (S3) line AM560-2 developed to minimise heterozygosity, and hence bypass some of the assembly challenges. Sequences were assembled *de novo* with Newbler (Fryslie, no date). The contigs of assembly (v4.1) span a total length of 419.5 Mbp while 12,977 scaffolds span 532.5 Mbp (N50 258.1 kbp). The assembly was estimated to cover 69% of the genome size containing 30,666 annotated genes which was 96% of the predicted protein-coding gene space (Prochnik *et al.*, 2012). The current version (v6.0) (Bredeson *et al.*, 2016)—referred to as a high-quality reference genome assembly—is based on Illumina reads (whole genome shotgun, mate pair, fosmid end, Hi-C) from AM560-2 and it was assembled *de novo* with Platanus (Kajitani *et al.*, 2014). The total assembly is 582.3 Mbp long with contig N50 of 27.7 kbp. Eighty-nine percent of the assembly is ordered into 18 chromosomal pseudomolecules with contig N50 of 29.8 kbp

and scaffold N50 of 28.4 Mbp. Covering approximately 75% of the estimated genome size, it is carrying 33,033 annotated protein-coding genes (annotation v6.1). Unassembled 200 Mbp of sequences are estimated to contain less than 1% of cassava genes. The same authors also estimated that the AM560-2 putative genome is homozygous over 93.6% of its length, in line with their expectations on S3 generation. There are ongoing efforts to further increase the completeness of the assembly with long PacBio reads (Rhoads and Au, 2015) for the repetitive regions (Goodstein, D. M. *et al.*, 2012; Phytozome, 2017).

At the same time, Wang *et al.* (2014) worked on draft genome assemblies of wild *Manihot esculenta* ssp. Flabellifolia (432 Mbp long assembly—58.2% of the estimated 742 Mbp long genome; contig N50 43 kbp; 34,483 annotated genes of which 33,310 protein-coding) and domesticated *Manihot esculenta* Crantz (495 Mbp long assembly—66.7% of the estimated 742 Mbp long genome; contig N50 19 kbp; 38,845 annotated genes of which 37,592 protein-coding) which were released together with transcriptome sequencing reads.

In 2019, Kuon *et al.* provided annotated haplotype-resolved genome assemblies of *Manihot esculenta* cultivar TME3 (spanning 634.1 Mbp of the estimated 765 Mbp (82.9%); 558 scaffolds, scaffold N50 2.25 Mbp; 33,853 annotated protein-coding genes) and cultivar 60444 (spanning 714.7 Mbp of the estimated 745 Mbp (95.9%); 552 scaffolds, scaffold N50 2.35 Mbp; 34,127 annotated protein-coding genes). Since cultivar TME3 is resistant to the cassava mosaic disease, an issue in Africa, and 60444 is disease-susceptible, the assemblies and annotations bear importance for studying the disease and developing disease-resistant cultivars.

1.4 Comparative genomics

The field of comparative genomics is concerned with answering biological questions by comparing genomic features of the organisms under study (Ellegren, 2008). Typical studies investigate similarities and differences between organisms, and the evolution of genomic features of interest (Hardison, 2003). Features include putative genomes, genes, their location on chromosomes, regulatory elements and other non-coding elements; these can be compared within and across species (Miller *et al.*, 2004; Ellegren, 2008; Alföldi and Lindblad-Toh, 2013). At a time of ever-increasing amounts of data, comparative methods help gain new insights, as well as to propagate existing knowledge, to the newly obtained data (Dunn and Munro, 2016).

1.4.1 Brief introduction to comparative genomics

The first comparative studies appeared in the 1980s and focused on comparisons between putative virus genomes (Toh, Hayashida and Miyata, 1983; Argos *et al.*, 1984; Haseloff *et al.*, 1984; Ahlquist *et al.*, 1985; McGeoch and Davison, 1986). The first comparison between genome sequences of cellular organisms was performed in the mid 1990s, with the completion of the first bacterial genome sequencing projects (Tatusov *et al.*, 1996). Since then, thousands of genome sequencing projects have been completed and the framework of comparative genomics provides means to understand and interpret the obtained data (Koonin and Galperin, 2010; Mukherjee *et al.*, 2019).

Central to the comparative genomics approach is the assumption that the genomes under study have evolved from a common ancestor (Ureta-Vidal, Ettwiller and Birney, 2003). Thus, the present genomes can be explained as a result of evolutionary processes that have acted over time since the present genomes started diverging from one another (Ureta-Vidal, Ettwiller and Birney, 2003). Mutational forces introduce random mutations in the genome, and based on their effect on the chromosome structure, they can

be classified into small-scale and large-scale mutations (Lodish *et al.*, 2000). Small-scale mutations affect up to 1,000 consecutive nucleotides and include substitutions, insertions and deletions (Lodish *et al.*, 2000; Wright, 2003). Thus, they might alter a function of a single gene, if any (Lodish *et al.*, 2000). Large-scale mutations include amplifications, deletions and changes in the location of larger genomic regions (Lodish *et al.*, 2000). Genes, chromosomes and even whole genomes can duplicate (Taylor and Raes, 2004). Deletions of larger genomic regions lead to, e.g. gene loss, loss of heterozygosity and formation of fusion genes (Griffiths, Miller and Suzuki, 2000; Albalat and Cañestro, 2016). DNA segments can reverse their orientation within a chromosome in the process of inversion or move to a different chromosome in the processes of translocation and crossover (Griffiths, Miller and Suzuki, 2000; Kirkpatrick, 2010; Roukos, Burman and Misteli, 2013). Large-scale mutations can affect multiple genes, thus, leading to phenotypic changes and sometimes even to speciation (Lodish *et al.*, 2000).

Mutations can be harmful (deleterious), neutral or beneficial for the organism (Ureta-Vidal, Ettwiller and Birney, 2003). Typically, harmful mutations are eliminated by negative selection, beneficial mutations become more common in a population thanks to the positive selection, and neutral mutations are not affected by selection (neutral selection) (Ureta-Vidal, Ettwiller and Birney, 2003). However, selection is not deterministic: both beneficial and neutral mutations can get lost, or deleterious mutations can get fixed, via genetic drift—an evolutionary process by which the frequencies of gene variants in a population change from generation to generation due to chance (Masel, 2011). Because of its stochastic nature, genetic drift particularly affects small populations or populations undergoing bottlenecks, e.g. due to dispersal or geographic isolation (Masel, 2011).

Speciation refers to the process by which two subpopulations become genetically isolated from one another, leading to the emergence of distinct species (Coyne and Allen Orr, 2004). Roughly speaking, this can happen through geographic isolation (allopatric speciation), partial geographic

isolation or while sharing the same habitat (sympatric speciation) (Butlin, Galindo and Grahame, 2008; Fitzpatrick, Fordyce and Gavrilets, 2009).

After speciation, it is generally assumed that each of the new species will evolve largely independently of the other (Wolf, Lindell and Backström, 2010). However, genetic material can be occasionally exchanged between species, by mechanisms such as horizontal gene transfer or by introgression (Gogarten, Gogarten and Olendzenski, 2009; Twyford and Ennos, 2012; Suarez-Gonzalez, Lexer and Cronk, 2018).

With the advent of DNA and protein sequencing from the second half of the twentieth century, it became apparent that, in vertebrate species and most multicellular eukaryotes, most molecular characters are evolving neutrally or near neutrally (Kimura, 1983). This implies that most substitutions observed in orthologous sequences between species are not the result of Darwinian selection, but rather the random process of random drift. This theory explains earlier observations of a “molecular clock”—a linear relationship between the rate of nucleotide substitutions between lineages and their divergence time as estimated from the fossil record (Zuckerkandl and Pauling, 1965). Later analyses, based on much larger datasets, have shown that this relationship is not always linear, with some sequences substantially departing from a molecular clock. Nevertheless, a substantial correlation between sequence similarity and divergence time can often be observed and thus the molecular clock often provides a helpful first approximation (Nei, Suzuki and Nozawa, 2010).

Sequences encoding or regulating shared features among organisms are typically evolutionary conserved, while sequences encoding or regulating the differences are typically divergent (Hardison, 2003). This reasoning allows the evolution of genomic sequences to be studied and, for example, predicting a core set of genes shared across species (even if the species are separated by more than 1 billion years of evolution), finding potentially functional DNA sequences (for species separated by 70-100 million years of divergence) or identifying potentially species-specific features (given the

dataset of species separated by ~5 million years of divergence) (Hardison, 2003)[27].

Comparisons between putative genomes usually start with pairwise comparisons of putative gene sequences in the dataset (Wei *et al.*, 2002). The aim of this initial comparison of two gene sequences is typically to determine whether they can be identified as putative homologs, i.e. sequences of genes believed to have a common ancestor. Homology is the key concept in comparative genomics (Dunn and Munro, 2016) and we discuss it in section 1.4.2 while section 1.4.3 touches upon homology inference. Two sequences can be compared by being aligned along their entire length—the technique called global alignment (using for example, Needleman-Wunsch algorithm (Needleman, 1970)) or only parts of the sequences can be investigated by exploring the local alignment (obtained by, e.g. Smith-Waterman algorithm (Smith and Waterman, 1981)).

Putative homologs can be classified further into putative paralogs (genes believed to have diverged through gene duplication) and putative orthologs (genes believed to have diverged through gene speciation) (Fitch, 1970). Paralogous genes are believed to drive function innovation while orthologous genes tend to have more similar biological function (Tatusov, Koonin and Lipman, 1997; Lynch and Conery, 2000; Altenhoff *et al.*, 2012; Gabaldón and Koonin, 2013). This has applications in protein function prediction (Gaudet *et al.*, 2011), inferring phylogenetic species trees and studying the evolution of gene families (Altenhoff and Dessimoz, 2012).

Relations between sequences can also be studied in a multiple-sequence setting, i.e. comparing three or more sequences at a time. A multiple sequence alignment (MSA) allows the identification of conserved regions among evolutionarily related sequences. The majority of tools for computing MSA of coding sequences implement a heuristic search called progressive alignment (first introduced by Hogeweg and Hesper (1984)) which starts by

[27] Estimated divergence distances

aligning two sequences and continues with successive alignment of the rest of the sequences, adding one by one. Widely cited tools employing variations of this approach include Clustal Omega (Sievers *et al.*, 2011), MAFFT (Katoh and Standley, 2013) and T-Coffee (Notredame, Higgins and Heringa, 2000). Iterative methods such as MUSCLE (Edgar, 2004a, 2004b) and PRRN (Gotoh, 1999) work similarly to progressive methods but realign all sequences with inclusion of each new sequence. This allows them to achieve higher accuracy. Some methods incorporate phylogeny information to avoid mistakes caused by structural matching and provide more accurate alignments for evolutionary inference. Examples include PAGAN (Löytynoja, Vilella and Goldman, 2012), PRANK (Löytynoja, 2014) and ProGraphMSA (Szalkowski, 2012). Some tools rely on hidden Markov models, e.g. POA (Lee, Grasso and Sharlow, 2002) and SAM (Hughey and Krogh, 1996), which allows them to generate all possible MSAs and assign a likelihood to each. There are also consensus methods which attempt to find an optimal MSA given multiple MSAs, for example, M-Coffee (Wallace *et al.*, 2006) and MergeAlign (Collingridge and Kelly, 2012).

Multiple sequence alignments are also used as input for phylogenetic tree building methods (Yang and Rannala, 2012). They are important for studying relationships among entities such as genes, organisms, species, clades (Soltis and Soltis, 2003). In a phylogenetic tree, leaves represent the data that is being compared while branches depict hypothetical relationships among them (Whelan, Liò and Goldman, 2001). A tree can be rooted or unrooted. In a rooted tree, an internal node represents the most recent common ancestor of its descendants and the root represents a common ancestor of all instances in the tree (Baldauf, 2003). Paths leading from the root to each of the leaves depict putative evolutionary paths (Whelan, Liò and Goldman, 2001). Different gene families evolve under different evolutionary processes (Ohta, 2000). Thus, the corresponding reconstructed gene trees will likely differ. An estimated rooted gene tree can be compared to an estimated rooted species tree in order to predict which evolutionary processes have acted on the genes—a process called phylogeny reconciliation (Doyon *et al.*, 2011). Depending on the complexity of the

reconciliation model, the inferred events assigned to internal nodes of the gene tree might include, for example, duplications and losses (DL models) or duplications, losses and horizontal gene transfers (DTL models) (Doyon *et al.*, 2011). Reconstructed gene trees can also help reconstructing a species tree and estimating divergence times between species (Delsuc, Brinkmann and Philippe, 2005; Yang and Rannala, 2012). Methods for building phylogenetic trees typically fall into one of the four groups: i) maximum parsimony methods (implemented in tools such as PAUP (Swofford, 2000), MEGA (Tamura *et al.*, 2011), TNT (Goloboff, Farris and Nixon, 2008)), ii) distance matrix methods (e.g. least squares (Cavalli-Sforza and Edwards, 1967), minimum evolution (Rzhetsky and Nei, 1992; Desper and Gascuel, 2002), neighbour joining (Saitou and Nei, 1987)), iii) Bayesian methods (implemented in, e.g. MrBayes (Huelsenbeck and Ronquist, 2001), BEAST (Drummond *et al.*, 2006)), and iv) maximum likelihood methods (implemented in, e.g. FastTree (Price, Dehal and Arkin, 2010), MOLPHY (Adachi and Hasegawa, 1996), PAUP* 4.0 (Swofford, 2000), PhyML (Guindon and Gascuel, 2003), PHYLIP (Felsenstein, 2005), RAxML (Alexandros Stamatakis, 2014)). A summary of their strengths and weaknesses can be found in the aforementioned review by Yang and Rannala (2012). Phylogeny reconstruction and applications is a field in itself and we refer readers to the textbooks such as Inferring phylogenies (Felsenstein, 2004) and Molecular Evolution: A Statistical Approach (Yang, 2014).

1.4.2 Homology

The first definition of homology in biological sciences was coined in 1843 when Richard Owen analysed the usage of the term “homolog” in anatomy and proposed the following definition in the Glossary of his Hunterian Lectures (Owen, 1843):

> “Homologue.”—The same organ in different animals under every variety of form and function.

The rather vague formulation did not mention common ancestry—understandably given that Darwin published On the Origin of Species in 1859 (Darwin, 1859). In his work from 1870, Edwin Ray Lankester acknowledges the problem of inferring the sameness of organs from Owen's definition and questions its investigation without the concept of evolution (Lankester, 1870). He goes on with proposing definitions in accordance with the theory of evolution and suggests splitting "homologues structures" into "homogenous" and "homoplastic structures":

> Structures which are genetically related, in so far as they have a single representative in a common ancestor, may be called *homogenous*.
>
> Homoplasy includes all cases of close resemblance of form which are not traceable to homogeny, all details of agreement not homogenous, in structures which are broadly homogenous, as well as in structures having no genetic affinity.

Although Lankester hoped to avoid confusion by introducing new terms, the meaning and usage of the terms "homology", "homogeny" and "homoplasy" kept evolving as can be seen in the overview of Haas and Simpson (1946). They then (re)defined "homology" as:

> a similarity between parts, organs, or structures of different organisms, attributable to common ancestry,

and "homoplasy"

> to comprise all evolutionary processes bringing about similarities between organisms or their parts, organs or structures, which are not due to common ancestry, but to independent acquisition of the similar characters.

In phylogenetic systematics, its founder Willi Hennig employed the same concept of homology (Hennig, 1965) which still prevails in modern systematics.

Homology can be defined at different levels or units. Owen and Lancaster thought of homology within the scopes of anatomy and morphology, speaking of homologous bones and organs (Owen, 1843; Lankester, 1870), while modern biology employs homologous thinking in genetics recognising

homologous nucleotides, amino acids, protein domains, proteins, genes, etc. (Fitch, 2000; Morrison, Morgan and Kelchner, 2015). In this thesis we often refer to *homologous genes* and *homologous proteins*. Homologous genes are genes that share a common ancestor. We also call this *gene-level homology* since a unit of homology is gene. Homologous genes code for homologous proteins, and vice versa, homologous proteins are coded by homologous genes. Thus, we refer to *protein-level homology*. A protein can have one or more protein domains which may or may not have the same evolutionary history, the latter being the consequence of gene fusion, domain insertion or domain deletion (Bornberg-Bauer *et al.*, 2005). For example, enterokinase contains a serine protease domain, two domains homologous to a domain of the low-density lipoprotein receptor, a domain homologous to the membrane-bound metalloproteases of renal glomeruli, a domain related to a protein of *Drosophila* dorsal-ventral patterning gene tolloid, and a domain homologous to cysteine-rich motifs found in macrophage scavenger receptor (Kitamoto *et al.*, 1994). Thus, the concept of homology may in some instances not apply at the protein-level while applying at a domain-level. In such cases, David M. Hillis proposes a term “partial homology” (Hillis, 1994), also recommended by Walter M. Fitch (Fitch, 2000).

It is important to distinguish “homology” from “predicted homology” (Patterson, 1988). “Homologous” means evolutionary related. Homologous structures that used to be identical evolve and diverge gradually reducing the similarity. However, the true sequence of changes is usually unknown. We are constrained to make observations and find supporting evidence for homology (often a certain level of similarity) studying mostly extant species. We might observe similarities which are not due to common ancestry but rather to a convergent evolution or random mutations, and it is not necessarily possible to tell them apart (Sanderson and Hufford, 1996). We might also fail to infer homologous relationships if the structures under examination have diverged beyond recognition of our methods (Fariselli *et al.*, 2007; Pearson, 2013). Hence, the inferred homologs are not necessarily true homologs, nor are all homologous relationships detectable. Throughout this thesis we use words *putative*, *predicted*, *inferred* in front of *homologs*

and *homologous relationships* to indicate that we speak about homology prediction.

Homology prediction often relies on sequence similarity (more in the following section 1.4.3) so we often speak of *putative homologous DNA/RNA sequences* or *putative homologous protein sequences*, or just *putative homologous sequences*. In cases of putative partial homology, the two corresponding protein sequences will contain sequences of domains which are putative homologous to each other, and those which are not. Even where homology holds on the gene- or protein-level, corresponding sequences of two homologs might not be of the same length as a consequence of insertions and deletions. Thus, the sequences will contain stretches of common sets of residues and those that are present only in one of them. Throughout this thesis where we make homology prediction, we refer to this fine-grained concept as *putative subsequence-level homology* or shorter *putative subsequence homology*.

1.4.3 Alignment methods for homology inference

Homologous genes—genes that originated from a single ancestral gene—evolve and diverge over time, leading to a gradual reduction in their similarity (Patterson, 1988). Looking at the sequencing data, if two sequences are more similar than expected by chance, i.e. share significant sequence similarity according to a chosen criteria, that could be explained as a consequence of common ancestry (the simplest explanation) or convergent evolution (a more complex explanation) (Pearson, 2013). Thus, many computational methods predicting homologous relationships consider higher-than-chance sequence similarity as an indication of homology (Chen *et al.*, 2016). However, detecting such statistically significant sequence similarity is not a trivial task, and the inability to do so does not imply non-homology (Pearson and Sierk, 2005; Pearson, 2013). Comparison of DNA sequences typically allows homology detection for genes which diverged up to 200-400 million years ago (Pearson, 2013). Since different codons can specify the

same amino acid, protein sequences are more conserved and can facilitate homology detection for genes with divergence time more than 2.5 billion years ago (Pearson, 2013). Protein sequence identity levels of 20-35% are often considered as a “twilight zone” for homology prediction (Vogt, Etzold and Argos, 1995)—yet protein sequences of homologous genes can share less than 20% identity (Pearson, 2013). Protein structure is thought to be often more conserved than protein sequence as it plays an important role in protein function (Illergård, Ardell and Elofsson, 2009). Thus, structure-based methods typically allow for better homology detection when sequences share low sequence identity (Fariselli *et al.*, 2007). As with sequences, it is important to be aware that significant structure similarity is just a signal of homology, and that the lack of detectable signal does not imply non-homology (Fariselli *et al.*, 2007). Despite potentially higher sensitivity of methods relying on protein structure, the majority of homology inference methods relies on protein sequence information as it is more convenient for modelling and requires less resources for practical applications (Chen *et al.*, 2016).

According to Chen *et al.* (2016), sequence-based homology inference methods can be roughly divided into three categories: i) alignment methods, ii) discriminative methods, and iii) ranking methods. Alignment methods look for indications of homology in sequence alignments or alignments of sequence representations (Wan and Xu, 2005). They are typically based on dynamic programming algorithms and their performance depends on the alignment algorithm and scoring function. The inference is made based on the alignment score which provides a better clue for homology than sequence identity and similarity measures (Chen *et al.*, 2016). Discriminative methods attempt to classify sequences into groups of putative homologs in a machine learning framework (Jaakkola, Diekhans and Haussler, 2000; Bernardes, Carbone and Zaverucha, 2011). Being trained on both positive and negative samples, they tend to make fewer false positive predictions than alignment methods. However, they rely on representing sequences as feature vectors which makes them challenging to develop and apply. Ranking methods approach homology inference as a ranking task and

provide a sorted list of putative homologs based on their spatial distance to the query in the feature space (Weston *et al.*, 2004; Liu, Chen and Wang, 2015; Chen, Liu and Huang, 2016). Taking advantages of both alignment and discriminative methods, ranking methods can achieve better predictive performance[28]. Discriminative and ranking methods can incorporate additional information beside the sequence itself which can further increase their predictive performance (e.g. physicochemical properties of amino acids), yet alignment methods tend to be more straightforward and less computationally demanding, thus more applicable to large datasets (Chen *et al.*, 2016). Methods which do not align sequences for the purposes of sequence comparison are referred to as alignment-free methods (Zielezinski *et al.*, 2019). Depending on the heuristics they employ, they can be much faster than the alignment-based approaches, and thus more scalable to large datasets. The most popular alignment-free methods for sequence comparison rely on *k*-mer counts (Zielezinski *et al.*, 2019).

Alignment methods can be classified further based on the alignment strategy into: i) sequence alignment methods, ii) profile alignment methods, and iii) methods relying on hidden Markov models (Chen *et al.*, 2016).

Methods relying on global or local pairwise sequence alignment make homology inference based on the pairwise sequence alignment score (Chen *et al.*, 2016). Global alignment (e.g. Needleman-Wunsch (Needleman, 1970)) is suitable for datasets containing sequences of similar length (Chen *et al.*, 2016). In other cases, especially when only parts of sequences are evolutionary conserved, local alignment (e.g. Smith-Waterman (Smith and Waterman, 1981)) is a better option (Chen *et al.*, 2016). Homology inference typically starts with all-against-all protein sequence comparison where all pairs of sequences, within and across species, are aligned and their alignment score is computed (Smith and Waterman, 1981; Altschul *et al.*,

[28] Some authors classify methods such as BLAST (Altschul *et al.*, 1990), FASTA (Pearson and Lipman, 1988; Pearson, 2000) and PSI-BLAST (Altschul *et al.*, 1997) into ranking methods (Liu, Chen and Wang, 2015) while others restrict to more complex methods which combine both alignment and discriminative approaches (Wan and Xu, 2005; Chen *et al.*, 2016).

1997). If the alignment score is above a certain threshold and alignment length constraints are satisfied (Dessimoz *et al.*, 2005; Roth, Gonnet and Dessimoz, 2008), a pair is called putative homologs. A standard implementation of Needleman-Wunsch or Smith-Waterman alignment algorithm takes time proportional to the product of the sequence lengths (Durbin *et al.*, 1998). Since all-against-all procedure scales quadratically to the number of sequences compared, it rapidly becomes very costly. To speed up the inference, methods adopt various heuristics which come at the cost of lower accuracy. For example, widely used approaches like BLAST (Altschul *et al.*, 1990) and FASTA (Pearson and Lipman, 1988; Pearson, 2000) do not guarantee finding optimal alignments (Fariselli *et al.*, 2007).

Profile alignment methods represent each putative gene family with a profile which can be used to find more family members and updated accordingly (Chen *et al.*, 2016). They tend to be more sensitive than sequence alignment methods because they calculate a profile based on position-specific information from the multiple sequence alignment (MSA) of a putative gene family (Gribskov, McLachlan and Eisenberg, 1987; Pearson, 2013). For the initial profile computation, a query sequence is searched against a set or database of sequences to detect sequences sharing statistically significant sequence identity (Chen *et al.*, 2016). Then an MSA is built from the query and its matches. Finally, a profile is computed from the MSA. A profile can be represented as a position-specific scoring matrix (PSSM) where the rows represent features (typically 20 rows for 20 amino acids) and columns represent residue positions in the chosen columns of the MSA (Fariselli *et al.*, 2007). Elements of the matrix contain or reflect the frequencies of each residue at each position in the alignment (Fariselli *et al.*, 2007)[29]. In generalised profiles, an additional row with a position-specific insertion/deletion penalty can be added (Gribskov, McLachlan and Eisenberg, 1987). Some methods compute a profile for each sequence in a database and query the sequence of interest against the database of profiles

[29] Some approaches generate a consensus sequence corresponding to a PSSM to aid visualisation of pairwise alignment of the profile and a query (Gribskov, McLachlan and Eisenberg, 1987).

(Schäffer *et al.*, 1999). Putative homology can also be inferred by comparing a profile to the database of profiles (Rychlewski *et al.*, 2000; Sadreyev and Grishin, 2003; Margelevičius and Venclovas, 2010). All three types of comparisons are typically performed using standard dynamic programming algorithms for pairwise sequence alignment or their variants (Gribskov, McLachlan and Eisenberg, 1987; Edgar and Sjölander, 2004). Methods relying on profile-to-profile comparisons tend to outperform profile-to-sequence and sequence-to-profile methods (Chen *et al.*, 2016). There are also profile-based methods which try to improve their performance with secondary structure information (Tomii and Akiyama, 2004; Chen and Kurgan, 2007; Kelley and Sternberg, 2009; Yang *et al.*, 2011; Gront *et al.*, 2012). Reliance on the MSA is also a disadvantage of the profile-based approaches (Pearson, 2013). The quality of the MSA will directly affect profile representation. Furthermore, inclusion of a non-homolog into a putative family can lead to more false positive predictions (Fariselli *et al.*, 2007; Pearson, 2013).

Profile hidden Markov models (HMMs) provide a probabilistic framework for homology inference (Durbin *et al.*, 1998). Like standard profiles, they capture position-specific information about the degree of conservation in the multiple sequence alignment (Krogh *et al.*, 1994). They can represent putative sequences and families, and homology search can be performed by comparing a sequence against a database of profile HMMs, a profile HMM against a database of sequences and a profile HMM against a database of profile HMMs, the last one being the most sensitive (Choo, Tong and Zhang, 2004; Wan and Xu, 2005; Remmert *et al.*, 2011). Unlike standard profiles, profile HMMs can model insertions and deletions (Durbin *et al.*, 1998). While profiles rely on observed frequencies and *ad hoc* scoring systems, profile HMMs have formal probabilistic basis and use statistical methods for parameter estimation (Eddy, 1998). As a consequence, a profile HMM built from a multiple sequence alignment of 10-20 sequences can have equivalent quality as a profile calculated from 40-50 aligned sequences (Rigden, 2017). In terms of homology prediction, methods based on profile HMMs are more

sensitive than sequence alignment and profile alignment methods (Choo, Tong and Zhang, 2004; Chen *et al.*, 2016).

A profile hidden Markov model (HMM) as a variant of HMMs for biological sequences has an underlying topology of a directed acyclic graph, with the exception of loops (a toy example depicted in Fig. 1.8) (Krogh *et al.*, 1994; Choo, Tong and Zhang, 2004). An HMM has a finite number of states which can generate observations (Rabiner and Juang, 1986). A profile HMM typically has three classes of states: match, insert and delete (Krogh *et al.*, 1994; Yoon, 2009). The sequence of states follows a Markov chain, thus the probability of being in a state depends only on the previous *n* states, where *n* is the order of the Markov chain. The probability of moving from one state to another or staying at the same state is called transition probability, and it has to be defined for all allowed transitions in the model. Each state generates a symbol independently of other states and is associated with an emission probability distribution. The emitted symbols are observed, and in contrast to Markov chains, it is not possible to tell which state emitted a symbol just by looking at the symbol. Thus, the name hidden Markov models. Two non-emitting dummy states can be added: the initial *Begin* state and the final *End* state (Krogh *et al.*, 1994). The *Begin* state allows the process to move to the first standard state according to a non-uniform probability distribution (Krogh *et al.*, 1994). The *End* state allows modelling the distribution of sequence lengths and defining a probability distribution on the sequence space (Krogh *et al.*, 1994; Durbin *et al.*, 1998). Starting from the *Begin* state, the process moves to the next state with a certain probability and a symbol is emitted according to the state's emission distribution. Then the process transitions to the next state, the state emits a symbol, and so on, until the process reaches the *End* state marking the end of the generated sequence. Each sequence generated by an HMM is independent of all others (Eddy, 1996). Given an HMM and an observed sequence of symbols, it is possible to calculate the probability that the sequence was generated by the HMM (forward algorithm) and find the most probable state path that generated the sequence (Viterbi algorithm) (Rabiner and Juang, 1986; Durbin *et al.*, 1998). When profile HMMs represent gene families, these probabilities help to detect putative

homologs (Krogh *et al.*, 1994; Krogh, 1998). Standard implementations of both algorithms for a profile HMM with *M* states have a runtime complexity *O(MN)* for a sequence of length *N*, i.e. the same as dynamic programming algorithms for pairwise sequence alignment (Eddy, 1998). The fact that different state paths can yield the same observed sequence (typically with different probabilities) contributes to the sensitivity of profile HMMs and helps them to outperform standard profiles for homology search (Chen *et al.*, 2016).

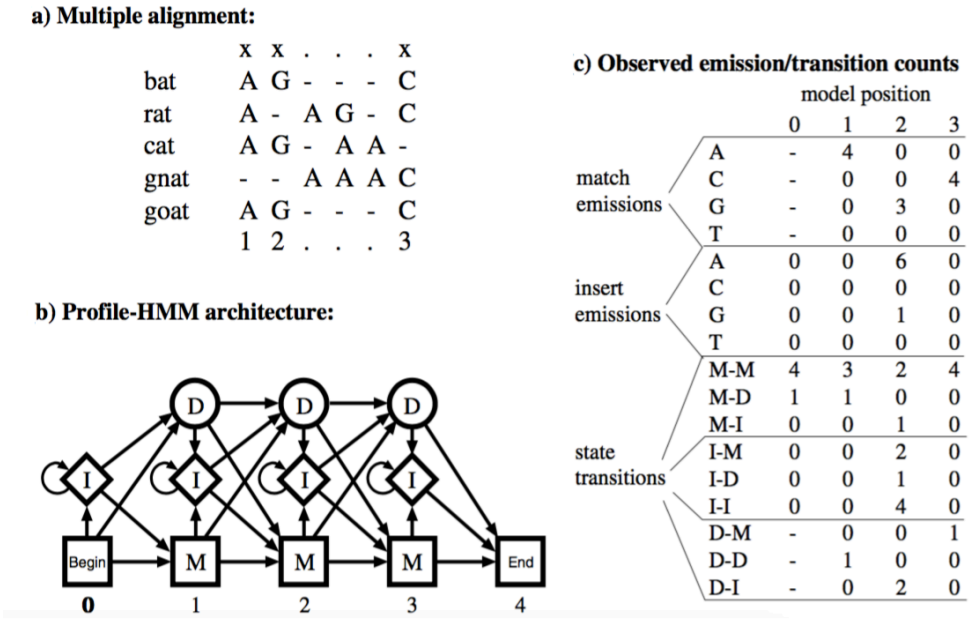


Figure 1.8: A toy example of a profile hidden Markov model (HMM) construction for a putative gene family (reproduced from: Durbin *et al.* (1998)[30]).

a) Multiple sequence alignment (MSA) of DNA sequences with *x*'s denoting columns assigned to match states (*M*) in the profile HMM. Column assignment can be done manually, heuristically or algorithmically (e.g. MAP match-insert assignment). The profile HMM assumes that the MSA is correct. b) Architecture of the profile HMM. *M*'s represent match states, *I*'s insert states and *D*'s delete states. Each match and insert state can emit one of the four nucleotide symbols (A, C, G, T) while delete states are "silent" and give rise to gaps (-). Arrows depict allowed transitions between different states or staying in the same state. A profile HMM of a gene family assumes that each homologous sequence is independently generated from the profile (equivalent to a star phylogeny with fixed branch lengths), and that each residue depends only on the underlying state (no correlation between residues) (Eddy and the HMMER development team, 2019). c) Counts of observed emissions from match and insertion states, and counts of

[30] From *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Richard Durbin, Sean R. Eddy, Anders Krogh and Graeme Mitchison, 1998, Cambridge University Press. This publication is in copyright. Subject to statutory exception and to the provision of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press. Reproduced with permission of Cambridge University Press through PLSclear. PLSclear Ref No 38316.

transitions between match, insert and delete states. Counts are used to estimate emission and transition probabilities of the model. Typically, weighting techniques are used to assign weights to sequences, e.g. lower weights to redundant sequences of closely related genes (Krogh and Mitchison, 1995; Karchin and Hughey, 1998).

In practice, runtime of (alignment-based) methods depends on the particular implementation and heuristics it employs. For example, according to the publication of a popular profile HMM-based method HMMER3 (Eddy, 2011), HMMER3 required a longer runtime on the benchmarking datasets than widely used profile-based PSI-BLAST (Altschul *et al.*, 1997) and sequence alignment-based NCBI BLAST (Camacho *et al.*, 2009), the latter two taking roughly the same time for inference. Sequence alignment methods FASTA (Pearson and Lipman, 1988; Pearson, 2000), SSEARCH (part of FASTA program family) and WU BLAST (Gish, 1996-2003) required even longer runtime. However, running a profile HMM-based method SAM (Karplus, Barrett and Hughey, 1998) took the longest time. In their publications on a profile HMM-based method HHblits, Remmert *et al.*, (2011) and Steinegger *et al.* (2019) reported experiments in which HHblits was faster than PSI-BLAST which was again faster than HMMER3.[31]

Faster homology methods with lower runtime complexity typically employ heuristics to avoid unnecessary pairwise sequence alignments. For example, CD-HIT (Li and Godzik, 2006), kClust (Hauser, Mayer and Söding, 2013), MMseqs (Hauser, Steinegger and Söding, 2016) and UCLUST (Edgar, 2010) divide input sequences into clusters (groups) of putative homologs quickly comparing a query and a cluster representative based on the number of common or high-scoring words (*k*-mers) in the sequences. Unfortunately, those algorithms are less sensitive and have problems when inferring distant relationships. Nevertheless, *k*-mer heuristics are promising. MMseqs2 (Steinegger and Söding, 2017), a more sensitive and faster successor of

[31] In terms of sensitivity, methods employing profile hidden Markov models outperformed profile-based PSI-BLAST. Methods relying on sequence alignment identified the fewest number of putative homologs.

MMseq, reached the sensitivity of BLAST (Altschul *et al.*, 1990) while being much faster (~36x) on the tested dataset. The tool also provides profile-to-sequence and sequence-to-profile searches which outperformed PSI-BLAST (Altschul *et al.*, 1997) in terms of sensitivity and runtime in the reported comparison (Steinegger and Söding, 2017). A recent study by Zielezinski *et al.* (2019) validated 24 tools for alignment-free sequence comparison. AFKS (Luczak, James and Girgis, 2019) and alfpy (Zielezinski *et al.*, 2017) demonstrated the best ability to distinguish between putative homologs and non-homologs. AFKS calculates different distance/dissimilarity measures between sequences based on *k*-mer counts while alfpy also calculates information-theoretic, graph-based and hybrid measures.

1.4.4 Comparative genomics methods to identify fragmentation in reconstructed protein-coding regions

Assembly and annotation of a newly sequenced genome or transcriptome are challenging tasks as described in sections 1.1.3 and 1.2. Published projects typically contain many fragmented sequences, a severe problem in plant research as discussed in section 1.3.1. Yet, fully assembled and correctly annotated protein-coding regions facilitate studying gene and species evolution, and provide insights into biological processes in the organism as explained throughout sections 1.2 and 1.4.1-1.4.3.

Taking into account the evolutionary conservation of genes and proteins (see section 1.4.1), already assembled putative gene and protein sequences can be used as guides in assembling homologous regions of newly sequenced species, for example, using reference-based assembly approaches as described earlier in section 1.1.2. Furthermore, putative homologous gene (protein) sequences can be used to guide scaffolding or, more generally, as a template for detecting fragments of the same gene (protein) model in a target assembly. Nevertheless, there is a possibility that a DNA sequence of a target or reference species was affected by mutations (as explained in section 1.4.1), and a putative sequence or annotated model from the

reference species could mislead resolving the problem at hand. Thus, the methods could introduce additional criteria and predictions could be further examined (computationally and/or experimentally) to acquire additional supporting evidence.

To our knowledge, there are four approaches that employ available putative homologous sequences to detect fragmentation in the target assembly and its annotation. They are described below.

ESPRIT (Dessimoz *et al.*, 2011) uses pairwise comparisons to identify non-overlapping pairs of putative protein-coding sequences that have a similar estimated evolutionary distance to putative homologs in other species. A pair of putative protein sequences corresponding to fragmented gene models is also required to consistently map together to multiple reference putative protein sequences. ESPRIT does not try to resolve cases where more than two fragments seem to belong to the same gene model or when fragments overlap (i.e. segments might come from the same haplotype, hence significantly overlap, but have not been assembled together due to uncertainty arising through polymorphism).

SWiPS (Li and Copley, 2013) is developed with the aim of exploiting putative orthology to guide scaffolding. It starts with identifying contigs that contain putative protein-coding exons and mapping them to the predicted orthologous reference protein sequences. Multiple contigs are allowed to map to the same protein sequence and, vice versa, a contig is allowed to map to multiple protein sequences. The similarity scores between every pair of a protein sequence and an assigned contig are then computed. These scores are used to greedily choose the best combinations of contigs to scaffold. After the scaffolding step is over, the algorithm inspects contigs containing multiple protein sequences and tries to use this information to connect scaffolds into super-scaffolds. The performance of the algorithm depends heavily on the ability to distinguish putative orthologs from putative paralogs, something which could be improved.

The Ensembl Compara pipeline (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020) infers as “gene_split” pairs of apparent paralogous sequences that lie within one megabase on the same strand of the same region of the assembly and do not overlap in the multiple sequence alignment of the putative gene family (EMBL-EBI, 2019b). Restricting these predictions to gene models annotated on the same contig greatly reduces the risk of false positive split gene model calling, but particularly for fragmented assemblies with many short contigs, this approach detects only a fraction of all splits. Furthermore, the pipeline cannot be easily run on custom genome data.

PEP_scaffolder (Zhu *et al.*, 2016) relies on high-identity matches of reference protein sequences to multiple contigs. Thus, like ESPRIT (Dessimoz *et al.*, 2011) and SWiPS (Li and Copley, 2013), the approach relies on pairwise alignments. Computationally particularly efficient, it also has the strength of considering a maximum intron length to avoid combining fragments that are unrealistically far apart.

Yet for all of these methods, the correct identification of split gene (protein) models heavily depends on their ability to distinguish fragments of the same putative gene (protein) sequence from fragments of paralogous sequences. Ensembl Compara (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020) and PEP_scaffolder (Zhu *et al.*, 2016) make no attempt to distinguish between the two. As for ESPRIT (Dessimoz *et al.*, 2011) and SWiPS (Li and Copley, 2013), although they attempt to identify fragments that match reference gene (protein) sequences consistently—either by requiring consistent estimated evolutionary distances to the reference for all fragments or by requiring consistent best matches for all fragments—these comparisons are inherently limited by the pairwise comparison setting, which loses out on phylogenetic information available in a multiple-sequence and tree setting.

1.5 Research questions and overview of the thesis

As we have seen throughout the previous sections, as the time and cost of sequencing decrease, the number of available putative genomes and transcriptomes rapidly increases. Yet the quality of the assemblies and annotations varies considerably and often remains poor, affecting downstream analyses (Schliesky *et al.*, 2012; Denton *et al.*, 2014). Many researchers are interested only in the protein-coding regions of a genome or transcriptome, or even just a single gene, and fragmented and incomplete assemblies and annotations might provide enough information for their particular studies. For example, a project aiming to identify evolutionary related functional genes shared across all mammals might benefit more from a large number of draft genome assemblies sampled across the class Mammalia rather than from a few high-quality assemblies (Margulies and Birney, 2008), particularly if the coding regions are sufficiently covered in the drafts (Thomma *et al.*, 2016). Fragmented and incomplete transcriptome assemblies have also proven useful in proteome analyses and pathway reconstructions (Schliesky *et al.*, 2012). On the other hand, many studies would benefit from assemblies of higher quality. For example, if a gene sequence is represented by fragments on separate contigs, each fragment might get annotated as a separate gene (as explained in sections 1.1.3 and 1.2). Homology might get inferred between the fragments and sequences from other species but the fragments might be mistaken as a pair of putative paralogs. Hence, such assemblies and annotations might not be optimal for studying lineage-specific deletions and gene family expansions (Margulies and Birney, 2008; Schliesky *et al.*, 2012). The fragmentation problem also affects non-coding regions (described in section 1.1.3), initially thought to be "junk", which are becoming increasingly important for studying life style, adaptability and evolution of organisms (Thomma *et al.*, 2016).

The more complex the genome, the more challenging sequencing, assembling and annotation are, as elaborated throughout the chapter. Multicellular species, especially plants with large and complicated genomic

sequences, will typically require more efforts to obtain complete and reliable assemblies (Chapman *et al.*, 2015) and consequently, annotation. Nonetheless, given the importance of plants for the human population and the need for higher yields and better plant protection, facing this challenge is necessary (International Wheat Genome Sequencing Consortium (IWGSC) *et al.*, 2018). Unfortunately, the majority of the available putative plant genomes are in a draft state and there is a lot of room for improvement (Claros *et al.*, 2012).

The potential improvements of current putative draft genomes and transcriptomes could be classified into three major groups. First, they could be resequenced using technologies which produce longer and/or more accurate reads. This could be done by using the existing methods or by developing new ones. This approach could be rather costly. Using the already existing approaches (e.g. Sanger (Sanger, Nicklen and Coulson, 1977), PacBio (Rhoads and Au, 2015), Nanopore (Feng *et al.*, 2015)) is more expensive than short-read sequencing. Developing new sequencing technologies could be expensive due to the costs of necessary wet-lab experiments. Second, there is an obvious lack of *de novo* algorithms that accommodate the complexities of plant genomes. Future developments could take them into account. This could be challenging especially if complexities are not well characterized and described in the literature, or if the solution is computationally costly. Finally, we could avoid making universal assumptions which hold across all species and work on a smaller scale within a framework of comparative genomics. We can exploit information from already available assemblies and resolve some ambiguities relying on evolutionary patterns across closely related species.

In this PhD project, we aimed to develop methods that detect fragments of the same gene model in an annotated assembly based on the information provided in already assembled and annotated putative homologous sequences from other species. If fragmented annotation is a consequence of a fragmented assembly, one could then improve the assembly as well. As more and more genome and transcriptome assemblies become available,

the average evolutionary distance between sequenced species is getting shorter which facilitates homology inference (Armstrong *et al.*, 2019), and hopefully makes the method more applicable and successful. Since homology inference requires many pairwise comparisons and corresponding computational costs rise rapidly when using putative plant genomes (or proteomes) due to their large size and complexity, we also worked on developing an efficient and sensitive approach to fast homology inference.

In the rest of the thesis, we provide a detailed description of the research contribution which can be divided into four research projects, and finish with concluding remarks.

In the following chapter, Chapter 2, we present a new approach to homology inference. The inference often starts with all-against-all pairwise putative protein sequence comparisons within and across species in a dataset of interest. It is a computationally intensive step in which many pairs do not turn out to be putative homologs. We present a clustering approach which speeds up homology inference by avoiding some of the unnecessary comparisons. Alongside the description of the algorithm, we present results on the real data and comparison to the established methods. The speedup is particularly relevant to large, complex and highly redundant putative plant genomes. Yet, as the approach was not parallelised at the time of working on the central aim of the thesis—developing and applying a method for detecting fragmented gene models—we did not use it in the subsequent projects. However, the new homology approach is a promising attempt. Thus, we provide an extensive discussion on its potential improvements in terms of recall and runtime, with a parallel implementation potentially having the highest impact on its wider applicability.

Having chosen a parallel mode of an existing homology inference method as a faster option to perform all-against-all comparisons, the rest of the work revolves around a pipeline for identifying fragmented gene models. Nevertheless, the all-against-all is the most time-intensive step in the pipeline, and putative homologs are the only source of information for

fragmentation inference. They thus underpin the work presented in Chapters 3-5.

In Chapter 3, we introduce two novel phylogenetic heuristics to infer non-overlapping or partially overlapping gene models that could be parts of the same, longer, gene model in a genome assembly of interest. One approach collapses branches with low SH-like support and the other makes inference based on the likelihood ratio value. We extensively validate the methods and analyse their performance varying input parameters. Having found a set of parameters which yields a reasonable number of reliable predictions, we apply the heuristics to the fragmented putative bread wheat proteome.

In Chapter 4, we explore the suitability of the heuristic approaches developed in Chapter 3 beyond putative bread wheat proteome. We apply them to the putative proteome of wild olive while providing practical guidelines to scrutinise their behaviour and results using no additional datasets.

In the last research project described in Chapter 5, we assemble and annotate a putative cassava transcriptome and use it as an input for our phylogenetic heuristics to survey their behaviour and performance on the transcriptomic data. The analysis reveals transcriptome-specific challenges and provides clues for future developments.

We conclude the thesis with a summary of the undertaken research projects and emphasise the importance of their outcomes.

Chapter 2: Speeding up homology inference

2.1 Introduction

As already mentioned in Chapter 1, the concept of homology plays a key role in computational biology studies such as protein function prediction, species tree reconstruction and gene family evolution.

Typically when predicting homologous relationships in a dataset, all possible pairs of putative gene or protein sequences within species and across species are inspected. This is computationally expensive, especially when the dataset comprises large putative plant genomes or proteomes. Since many pairs do not turn out to be putative homologs, computational costs could be reduced by avoiding unnecessary comparisons. Indeed, instead of performing all-against-all on the whole dataset, we could do a two step procedure: i) cluster the data, and ii) perform the all-against-all only within clusters. The challenge lies in the clustering step which should be fast and provide a baseline for accurate homology inference.

In this chapter, we describe our new approach to fast homology inference comprised of clustering followed by all-against-all comparisons within clusters. It was developed with the aim of speeding up homology inference adopted in OMA standalone software tool (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019). Our approach distinguishes itself from most clustering methods in that it relies on clustering based on full dynamic programming algorithm of Smith and Waterman (Smith and Waterman, 1981) for pairwise alignment comparison with cluster representatives and allows sequences to match to several clusters while attempting to exploit the transitive property of homology. The algorithm does not employ any heuristics on *k*-mer analysis to further speed up the inference process. The clustering approach could also be adapted for or incorporated into other homology inference pipelines relying on all-against-all comparisons

within a dataset. For example, it could be used instead of BLAST (Camacho *et al.*, 2009) all-against-all in the Ensembl pipeline (Zerbino *et al.*, 2018) for modeling gene families not described in Ensembl's library of HMM profiles (EMBL-EBI, 2019a; EMBL-EBI, 2019c). We also provide results of the comparisons with the current OMA approach, kClust (Hauser, Mayer and Söding, 2013) and UCLUST (Edgar, 2010). Finally, we discuss performance of the approach, its potential beyond the current framework and ways for improvement.

2.2 Methods

This section describes algorithm development of a model that takes into account complexities of real biological data. It also contains description of comparisons with the current OMA all-against-all approach (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019) and fast *k*-mer methods, namely kClust (Hauser, Mayer and Söding, 2013) and UCLUST (Edgar, 2010), and provides information on the empirical datasets used throughout the study.

2.2.1 Building clusters of putative homologous sequences in an ideal theoretical framework

Homology is an equivalence relation. In particular for gene-level homology defined on the set of all genes, that means that, for any three genes A, B and C, it is:
i) *reflexive*: gene A is homologous to gene A,
ii) *symmetric*: gene A is homologous to gene B if and only if gene B is homologous to gene A,
iii) *transitive*: if gene A is homologous to gene B, and gene B is homologous to gene C, then genes A and C are homologous.
As an equivalence relation, it partitions the set of all genes into disjoint equivalence classes—sets of homologous genes or homologous clusters—

where all genes within a homologous cluster are homologous to each other and there is no other gene outside the cluster that is homologous to any of them. A homology inference method which accurately identifies all homologous and non-homologous relationships among genes is, thus, an equivalence relation[32].

A corollary of transitive property of homology is that if genes A and B are homologous, and gene A is not homologous to gene C, then genes B and C are not homologous.

Proof of corollary: Let S1 denote “Gene A is homologous to gene B”, S2 “Gene B is homologous to gene C” and S3 “Gene A is homologous to gene C”. The transitivity of homology states that $S1 \wedge S2 \Rightarrow S3$. By material implication, that is equivalent to $\neg (S1 \wedge S2) \vee S3$. De Morgan’s law yields the equivalent expression $(\neg S1 \vee \neg S2) \vee S3$ which is, due to the associativity and commutativity of disjunction, equivalent to $(\neg S1 \vee S3) \vee \neg S2$. Using De Morgan’s law again, the expression transforms to the equivalent $\neg (S1 \wedge \neg S3) \vee \neg S2$. Finally, by material implication it is equivalent to the expression $S1 \wedge \neg S3 \Rightarrow \neg S2$.

■

The corollary can be exploited in methods which aim to provide evidence for homology from putative gene or corresponding putative protein sequences. More precisely, it is applicable in partitioning a set of putative gene sequences or a set of putative protein sequences into clusters of putative homologous sequences given that the method is capable of detecting every pair of sequences derived from homologous genes (proteins), even when the genes are evolutionary distant. In particular, once the pairs of sequences A and B, and A and C are inspected for evidence of homology, there is no need to look for the evidence between sequences B and C. Furthermore, for a cluster of putative homologous sequences, one putative gene (protein)

[32] It is probably impossible to develop such a method due to unknown and complex mechanisms of evolution.

sequence can act as a representative for the whole cluster. If a new sequence is putative homologous to the representative, due to transitivity of putative homology it is putative homologous to all other sequences in the cluster. Similarly, if a sequence is not putative homologous to the representative, it is not putative homologous to any sequence in the cluster.

2.2.2 Inferring clusters of putative homologs on the available real data

In practice, it can be very difficult to find evidence for homology. At a sequence level, evidence for homology can be found only in the common set of residues of the examined putative sequences. We do not know the true evolutionary history of genes and homology inference methods may have unrealistic assumptions, e.g. a chosen score or probability threshold that a pair of putative sequences should satisfy in order to be called putative homologous (more in section 1.4.3). Consequently, such a homology inference method defined as a relation on a set of available putative gene or protein sequences might not necessarily hold properties of an equivalence relation.

In this project, we aim to construct a method that takes a set of putative protein sequences and classifies them into subsets of putative homologs. To reduce the number of comparisons, we try to use transitivity of homology and its corollary (see section 2.2.1), and choose a representative sequence for each cluster. Yet, we face two major problems. First, it can be challenging to find evidence for homology in putative sequences coming from very distant genes, i.e. we can easily miss homologous relationships when a cluster representative and a query are putative protein sequences of distant genes. Second, due to insertions, deletions, gene fusions and fissions, at a sequence level transitivity holds only on the common set of residues. As a consequence, we will miss a pair of putative homologs if the representative does not cover common residues of a sequence under inspection and its putative homolog within the set (Fig. 2.1). Also, a sequence added to the

cluster based on its common residues with the representative might not have common residues with all other sequences in the cluster.

To bypass these problems, we introduce four modifications to the described algorithm. First, we use more than one representative per cluster to improve inference when the putative sequences are so diverse that one sequence is not a good representation of the whole cluster. Second, we allow putative sequences to be assigned to more than one cluster. This is beneficial for clusters which are fragmented due to sequences coming from highly diverged genes, and for multi-domain protein sequences. Third, we consider putative subsequence-level homology (the concept described in the end of section 1.4.2). In our algorithm, if an entire length of a putative sequence (minus 20 amino acid residues tolerance) is not covered by the representatives of the assigned clusters, the sequence also founds a new cluster and becomes its representative. Finally, once the clusters are built, we perform all-against-all comparisons within each cluster to eliminate false putative homology calls that can be detected with our criteria.

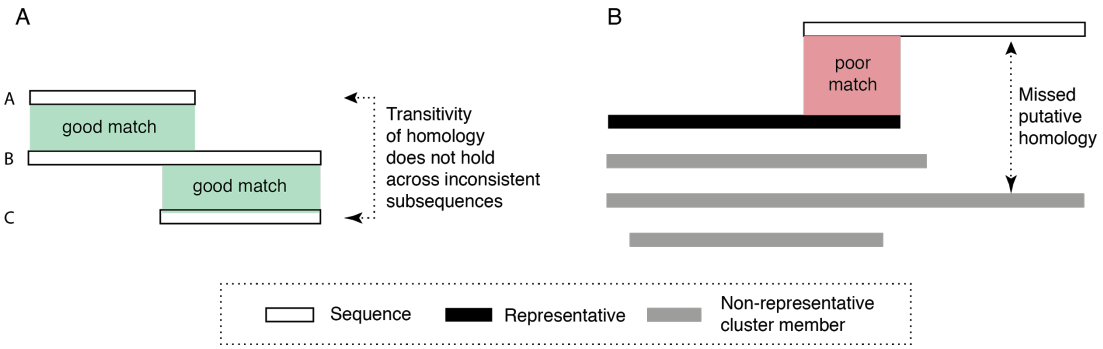


Figure 2.1: Diagram of potential problems with exploiting transitive property of homology in inference on the real data.

a) The residues involved are not consistent, and after finding indications that putative sequences A and B come from homologous genes, but not finding indications and A and C come from homologous genes, one would conclude that putative sequences B and C do not come from homologous genes either. b) The representative sequence does not cover significant regions of two mutually putative homologs and the relationship is missed.

The pseudocode of our final cluster building procedure is shown in Figure 2.2. A set *Proteomes* comprises input putative proteomes ordered by their number of putative sequences in descending order while putative sequences within putative proteomes keep the same ordering as in the original database file. Starting with the largest putative proteome, we process sequence by sequence. At the onset of the process, there are no clusters and the first putative sequence founds a cluster and becomes its representative. Every subsequent sequence is aligned with the representatives of the current cluster(s). To compute *alignment-score* between two sequences, we run Smith-Waterman pairwise alignment (Smith and Waterman, 1981) implemented by Szalkowski *et al.* (2008) using 224 GCB scoring matrix (Gonnet, Cohen and Benner, 1992). If the score is above the minimum threshold *T* of 135.75, the sequence is added to the cluster. Furthermore, if the number of cluster representatives in the particular cluster is below the maximum allowed (*L*), the newly added sequence becomes a representative. We also keep track of how much the putative sequence is covered by the representatives of the assigned clusters. We do not introduce any restrictions

on the size of the clusters nor on the number of clusters that a putative sequence is assigned to. After exhaustive search through all cluster representatives, we assess whether the putative sequence was added to one or multiple clusters. If the full length of the putative sequence (minus a tolerance *C* of 20 amino acids) is not covered by the clusters to which the sequence was added, an additional cluster is created with the sequence as a representative. The same applies if the putative sequence was not assigned to any clusters.

```
Input: Proteomes, thresholds T=135.75 (minimum score), L (number of representatives per cluster) and
       C=20 (minimum coverage)

Clusters ← ∅
for each proteome in Proteomes do
    for each sequence in proteome do
        if Clusters = ∅ then
            initialise newCluster
            newCluster ← sequence
            newCluster.representatives ← sequence
            Clusters.append(newCluster)
        else
            for each cluster in Clusters do
                for each representative in cluster.representatives do
                    if alignment-score(sequence, representative) > T then
                        cluster.append(sequence)
                        if |cluster.representatives| < L then
                            cluster.representatives ← sequence
                        end if
                    end if
                end for
            end for
        end if
        if sequence was assigned to at least one cluster then
            coverages ← ∅
            for each representative of the assigned clusters do
                coverages.append(coverage(sequence, representative))
            end for
        end if
        if sequence was not assigned to any cluster or sumlengths(coverages.uncovered_parts()) > C then
            initialise newCluster
            newCluster ← sequence
            newCluster.representatives ← sequence
            Clusters.append(newCluster)
        end if
    end for
end for

Output: a set of homologous clusters Clusters
```

Figure 2.2: Pseudocode of the new clustering approach.

Two details can be noticed in the implementation description—ordering putative proteomes according to the number of putative sequences and alignment score threshold *T=135.75*. They emerged from our empirical analysis of various clustering strategies on a small dataset comprised of 5 random putative bacterial proteomes (Tables A.1-A.2, Fig. A.1, explanation in section A.1.1). Although runtime was a part of the investigation, the analysis focused on finding a strategy that maximises recall.

2.2.3 Computing all-against-all within each cluster

After assigning all putative sequences to clusters, we run all-against-all procedure within each cluster keeping the same criteria as in the global all-against-all which we are trying to speedup. This way we ensure that resulting pairs fulfill the same criteria as the full all-against-all, and obtain pairwise scores often needed for downstream analyses. A side-by-side comparison of the new and the existing approach is shown in Figure 2.3.

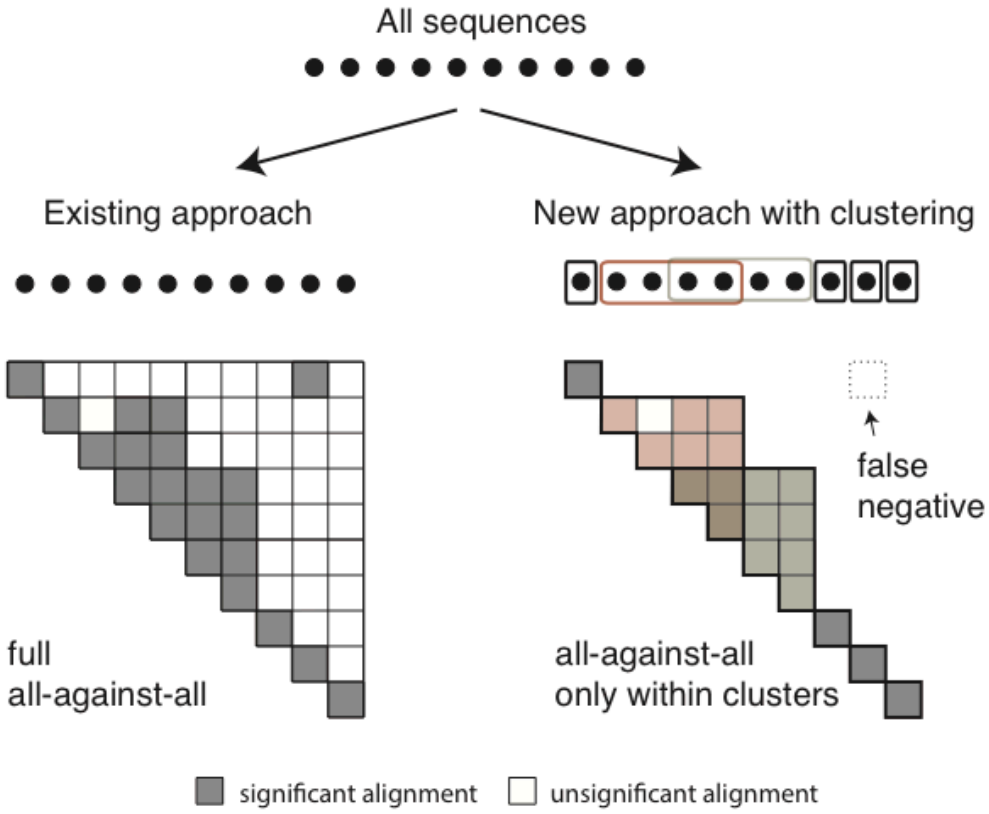


Figure 2.3: Comparison between the current all-against-all approach (left) and the new approach (right).

In the proposed approach, the putative sequences are first clustered and then the all-against-all is run only within clusters. Due to the clustering, the overall number of computations is considerably reduced, but some putative homologous relationships inferred by the full all-against-all approach can be missed using the new approach.

2.2.4 Runtime complexity of the new approach

If all putative homologous relationships among n putative sequences can be determined this way, the approach requires roughly nk comparisons to classify n sequences into $k \leq n$ clusters and roughly $\frac{1}{2}\frac{n}{k}\left(\frac{n}{k}-1\right)$ comparisons within each cluster—overall roughly $nk + \frac{n(n-k)}{2k^2}$ comparisons. Hence, the runtime complexity of the procedure is still $O(n^2)$ as in the case of all-against-

all approaches requiring $\frac{1}{2}n(n-1)$ comparisons. However, the new algorithm is more efficient than the $O(n^2)$ all-against-all approaches when $nk + \frac{n(n-k)}{2k^2} < \frac{n(n-1)}{2}$, i.e. when $\frac{k(2k-1)}{k-1} < n, k > 1$. The smaller the number of clusters k, the more efficient new approach will be. Particular speedup can be expected on the datasets where $k << n$.

2.2.5 Comparison with full all-against-all and other methods

To evaluate the new approach, we compared its performance to the baseline all-against-all approach used in OMA database algorithm (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019) and fast *k*-mer clustering approaches kClust (Hauser, Mayer and Söding, 2013) and UCLUST (Edgar, 2010).

We ran OMA algorithm (Roth, Gonnet and Dessimoz, 2008) with default parameters including: `MinScore := 181` (pairwise alignment score threshold), `LengthTol := 0.61` (length tolerance ratio[33]) and `MinSeqLen := 50` (sequence length threshold in amino acids). They were determined suitable for the real data by Roth, Gonnet and Dessimoz (2008). Computations were performed on a cluster.

We ran the *k*-mer algorithms kClust (Hauser, Mayer and Söding, 2013) and UCLUST (Edgar, 2010) with parameters listed in Table 2.1 and considered only putative sequences which were at least 50 amino acids long to keep consistency with other prediction methods in the study. Notably, we set a sequence identity threshold between a query and a cluster representative to 30% in all variants of the tested *k*-mer approaches which limited the number of putative homologous pairs that could be detected. CD-HIT (Li and Godzik, 2006), another widely used *k*-mer approach, limits sequence identity to the minimum of 40% and is reportedly less sensitive than kClust and UCLUST

[33] If the length of the effective alignment of putative sequences s_1 and s_2 is less than $LengthTol*min(length(s_1),length(s_2))$, the pair (s_1, s_2) is discarded.

using the same identity threshold on single-domain and multidomain protein datasets (Hauser, Mayer and Söding, 2013). Therefore, it was omitted from this comparison.

	kClust	UCLUST
Run 1	`-s 1.12, -c 0.8`	`-cluster_fast, -id 0.3`
Run 2	`-s 1.12, -c 0.5`	`-cluster_fast, -id 0.3, -target_cov 0.5, -maxaccepts 0, -maxrejects 0`
Equipment	`Dell R910 (Intel Xeon E7-8837, 2.66 GHz), 32 cores, 1TB RAM; used a single core`	`MacBook Air(Intel Core i7, 1.7GHz), dual-core, 8GB RAM`

Table 2.1: Parameters and equipment used for comparison with the *k*-mer approaches.

kClust (Hauser, Mayer and Söding, 2013) parameter `-s` stands for a clustering threshold. We used the default value of 1.12 which translates into ~30% sequence identity between a query and a cluster representative. Parameter `-c` represents alignment coverage of the longer putative sequence expressed as a fraction of its length. The default value was set to 0.8. In UCLUST (Edgar, 2010), `-cluster_fast` runs the algorithm variant optimised for speed and requires setting an `-id` parameter—the minimum sequence identity of a hit expressed as a fraction of columns in the alignment. We chose 0.3, i.e. 30%. In the subsequent run, we additionally required that 50% of a target sequence is covered by a query sequence when they are aligned (`-target_cov 0.5`), and with `-maxaccepts 0, -maxrejects 0` turned on an exhaustive putative homology search inspecting all possibilities (by default, the algorithm stops as soon as a query sequence is assigned to a cluster and it inspects up to eight most likely clusters per query).

2.2.6 Datasets

We used three datasets for evaluation. Bacteria dataset comprised 14 putative proteomes, fungi dataset 12 putative proteomes, and diverse dataset consisted of putative proteomes of one bacterium, one fungus, one plant, one protist and two vertebrates. All the data was exported from the OMA database (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018), March 2014 release. More details on all datasets as well as the distribution of sequence lengths and the distribution of estimated evolutionary distances of OMA putative homologs for bacteria and fungi datasets are provided in Appendix A (Tables A.4-A.6, Fig. A.3-A.4).

To inspect the scaling performance of the proposed approach in the number of input sequences, we ran its variants on subsets of different sizes. To get the subsets, we first sorted all putative proteomes in a dataset according to their number of putative sequences. For the first subset we took the central two putative proteomes from the list, for the second subset central four, and so on until all putative proteomes were included. The same procedure was done on both bacteria and fungi dataset. The scaling behaviour of the algorithm was also inspected on the diverse dataset where we performed a single run on the whole dataset.

2.3 Results

We implemented four variants of the algorithm—having one or three cluster representatives in combination with either taking or not taking into account the putative subsequence-level homology (see section 1.4.2 for definition). We were interested in both runtime and the ability to identify putative homologs. The latter was tested by taking OMA putative homologs (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019) as a reference set.

2.3.1 Results in a nutshell: 2x-9x speedup and >99% accuracy

In comparison to full all-against-all OMA procedure (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019) on bacteria and fungi datasets, all tested cluster variants showed decrease in runtime yielding speedup ranging from ~2x to ~9x (Fig. 2.4). The strongest speedup was achieved by the variant with one cluster representative which does not take into account putative subsequence-level homology (definition in the end of section 1.4.2), as expected—this variant requires the least number of comparisons to assign putative sequences to clusters and it assigns each putative sequence to exactly one cluster. Ignoring putative subsequence-level homology seems to contribute less to the overall speedup than reducing the number of cluster representatives.

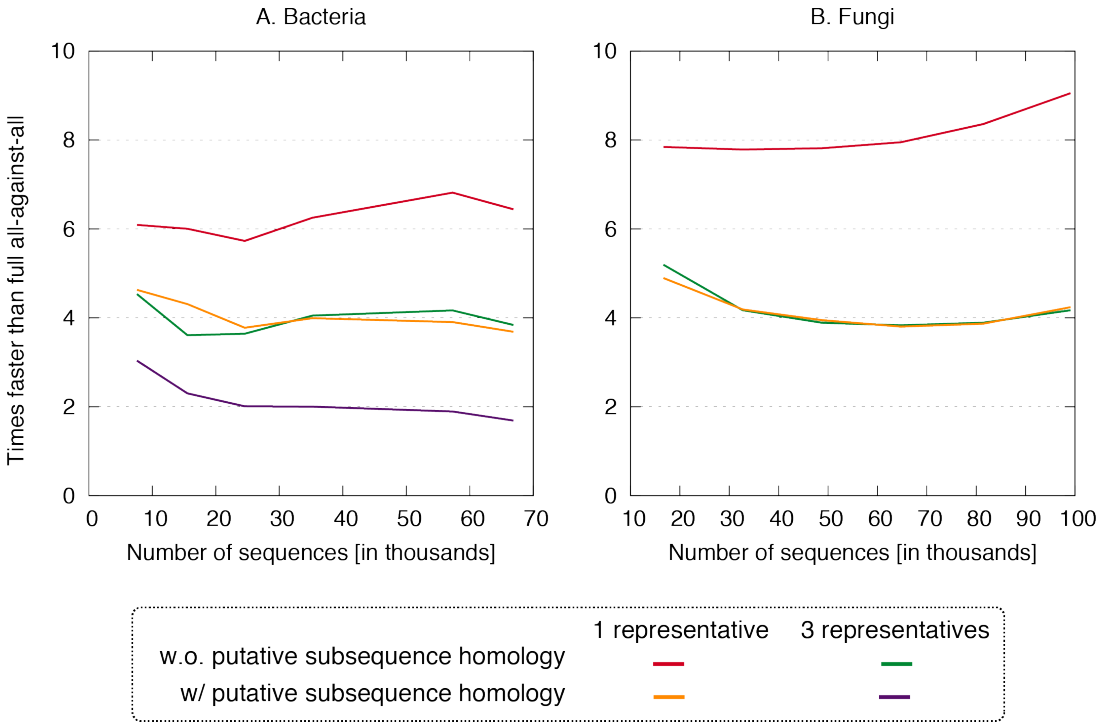


Figure 2.4: Speedup achieved by the new method.

The proposed approach is 2-9x faster, depending on datasets and variants. On the fungi dataset, only the three fastest variants were computed.

In terms of recall, all tested variants were able to identify >90% of putative homologous pairs identified by the current OMA all-against-all approach (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019) on bacteria and fungi datasets (Fig. 2.5). Taking into account putative subsequence-level homology (definition in section 1.4.2) reduced the number of missed putative homologs as anticipated. Furthermore, such variants achieved recall values of >99.6%. Unsurprisingly, the recall increased when the number of cluster representatives increased from one to three. Since recall values for all variants were consistent across bacteria and fungi subsets of different sizes, we may expect to observe similar recall values across other datasets.

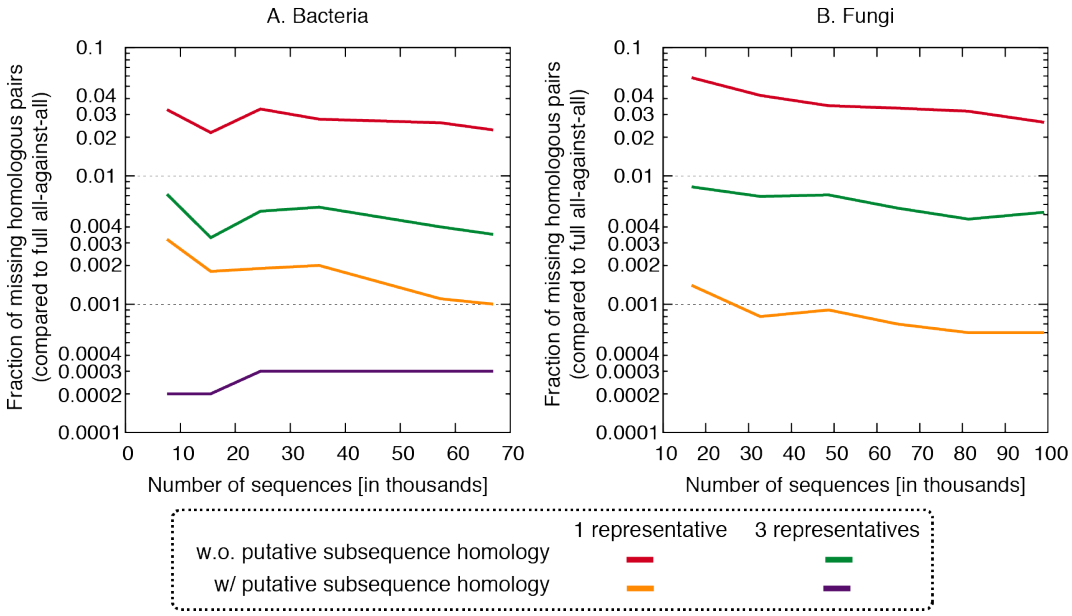


Figure 2.5: Fraction of OMA putative homologous pairs (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019) which are not identified with the new approach.

The new approach misses 0.4-6% of the pairs from the full all-against-all in its simple variant (1 and 3 representatives), and 0.01-0.3% taking into account putative subsequence homology.

Putting runtime and accuracy together, the best results were obtained by taking into account putative subsequence-level homology (concept defined in section 1.4.2) using a single representative sequence, which achieved a ~4x speedup while maintaining >99.6% recall in both datasets.

2.3.2 Robust to large putative proteomes and multidomain proteins

To investigate robustness of the new approach to large putative proteome sizes and putative multidomain proteins, we took as input the whole diverse dataset (two vertebrates, one plant, three unicellular organisms) and ran the algorithm variant with one representative taking into account putative subsequence-level homology (see section 1.4.2 for concept definition). It was 12.05 times faster than the OMA approach (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019), and the recall was

99.94%. This suggests that the approach is indeed robust to large putative proteomes and numerous putative multidomain proteins.

2.3.3 Tendency to miss lower-scoring putative homologous pairs, lower fraction of missed pairs in larger putative families

Taking OMA all-against-all output (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019) as a baseline, we investigated the distribution of alignment scores of pairs missed by the new approach (Fig. 2.6). All four variants on full bacteria and fungi datasets tend to miss pairs with lower scores, i.e. pairs with alignment score closer to the baseline threshold of 181. This is the most obvious in case of the variant which employs one cluster representative and ignores the putative subsequence level homology (concept explained in section 1.4.2), where the distribution is heavily skewed toward lower-scoring pairs. Generally speaking, such outcome is favourable since pairs with lower scores are challenging to identify reliably and the downstream analyses take that into account. Comparing the fraction of missing pairs with one versus three cluster representatives, we observed that the decrease in the fraction of missing pairs occurred mainly on the lower-scoring pairs.

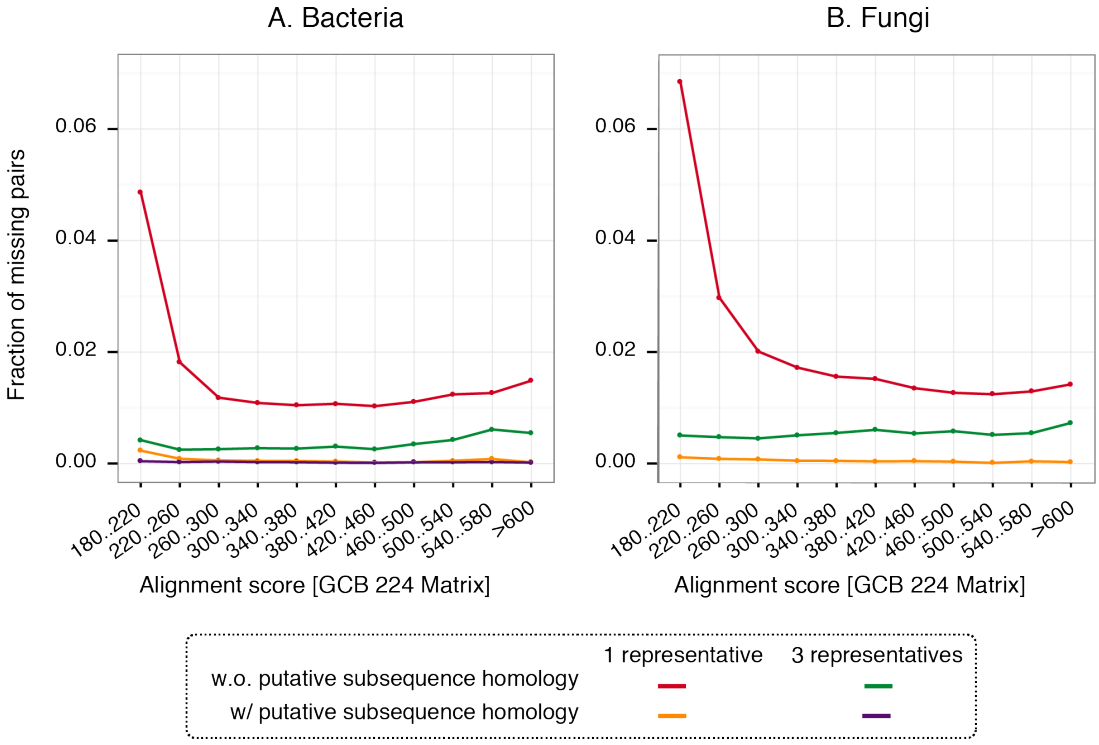


Figure 2.6: Distribution of alignments scores of missing pairs compared with all pairs identified by full OMA all-against-all approach (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019).

Missed putative relationships are heavily skewed toward lower-scoring pairs. Please note that only in the clustering step the alignment score threshold is 135.75. Afterwards when computing all-against-all within clusters, a pair is called putative homologs only if its alignment score is at least 181—the same criterion as in the full OMA all-against-all approach.

In addition, we investigated the fraction of missing putative homologous relationships per protein sequence. We predicted homologous relationships on the whole bacteria and fungi datasets using the algorithm variant with one representative and putative subsequence homology (explained in section 1.4.2). The proportion of missing homologs is very low regardless of the putative size of protein families (measured in terms of number of putative homologous relationships), with a trend to be lower in larger families (Fig. 2.7).

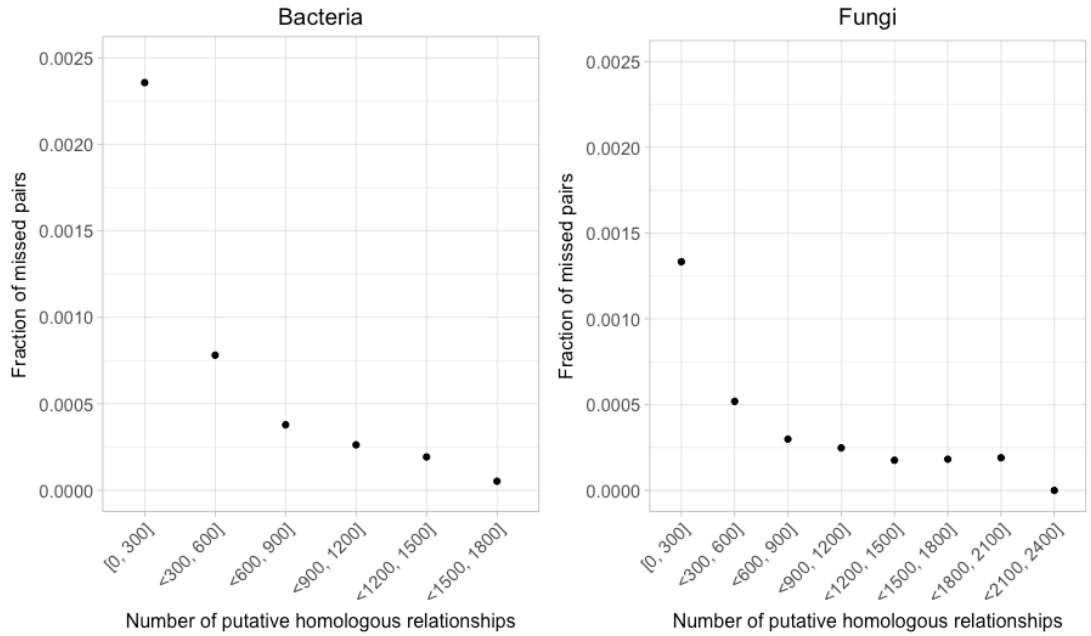


Figure 2.7: Fraction of missing putative homologous relationships when putative protein sequences were grouped according to the number of putative homologous relationships, for the full bacteria and fungi datasets with one representative and putative subsequence homology.

To illustrate the nature of missing putative homologs, we provide two detailed descriptions of high-scoring putative homologous pairs that are missed by the new approach (1 representative, putative subsequence homology (see section 1.4.2); see section A.3). In both cases, one putative sequence is added to a cluster with an alignment score slightly above the threshold while the other is not added due to a score just below the threshold.

2.3.4 Going downstream: more than 99% putative orthologs recovered

To assess the impact of the new clustering approach on orthology inference, i.e. how missed putative homologous relationships translate into missed putative orthologous relationships, we ran the OMA standalone software (Roth, Gonnet and Dessimoz, 2008; Train *et al.*, 2017) using our best new variant (1 representative, with putative subsequence homology (see section 1.4.2)), and compared predicted orthologs to the orthologs inferred by OMA

following the full all-against-all. On the largest bacteria and fungi dataset, the proportion of putative orthologs that were recovered was 99.71% and 99.87% respectively. This is slightly lower than the proportion of recovered putative homologs (99.9% and 99.94%), but remains very high.

2.3.5 Datasets too small to make conclusions on asymptotic behaviour of the number of clusters

For any given taxonomic range, adding new putative proteomes and the corresponding new putative sequences can only increase the number of clusters. However, as the number of proteomes grows, we can expect that an increasing proportion of the sequences will fall into one of the existing clusters. Thus, we should see a tapering in the number of clusters as a function of the number of putative sequences, which would be favourable in terms of runtime of the algorithm. In our datasets, we could not observe such tapering, and instead the growth in cluster numbers was broadly linear (Fig. 2.8 for bacteria, Fig. 2.9 for fungi subsets). This suggests that 12-14 putative proteomes are too few to discern the additional asymptotic benefits of our new approach.

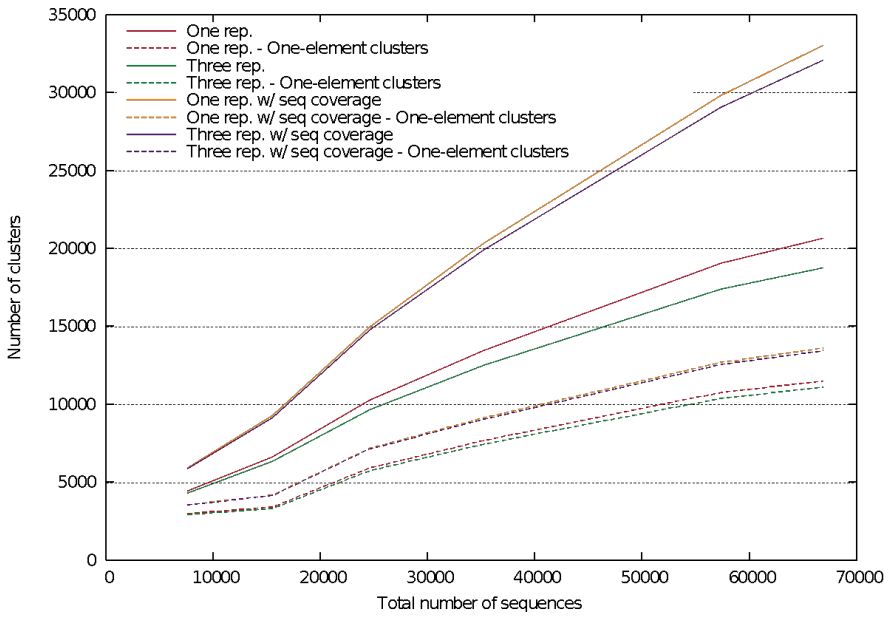


Figure 2.8: Growth of number of clusters on bacteria dataset.

No tapering is observed in the growth in the number of clusters generated by the new method.

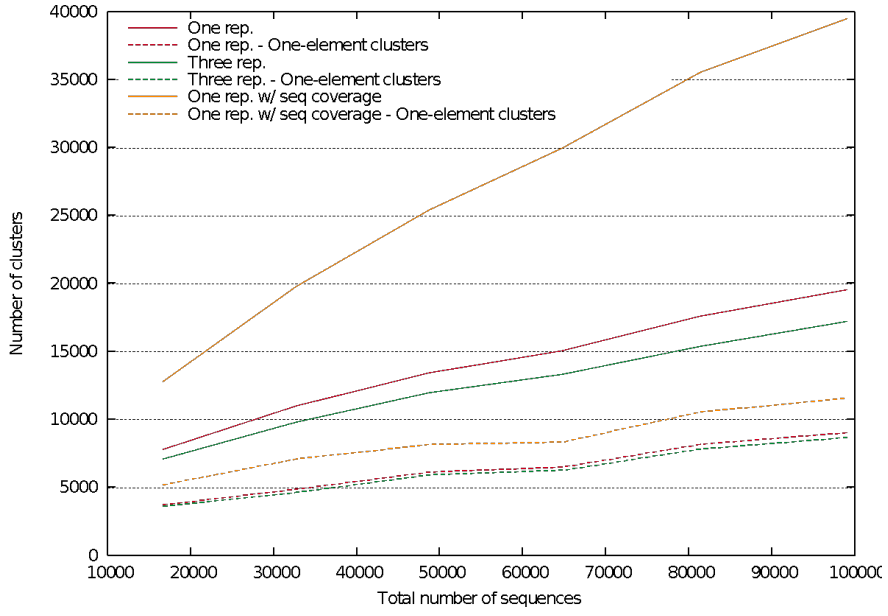


Figure 2.9: Growth of number of clusters on fungi dataset.

No tapering is observed in the growth in the number of clusters generated by the new method.

2.3.6 Skewness toward small cluster sizes

To gain insights into cluster sizes, we investigated clusters obtained on full bacteria and fungi datasets computed, again, by the algorithm variant with one representative and putative subsequence homology (concept defined in section 1.4.2). The distribution of cluster size is heavily skewed toward very small clusters (Fig. 2.10). The large overlap among numerous clusters and the existence of many putative sequences included in multiple clusters (Fig. 2.11) suggests improvement potential by merging some of the clusters (see also section 2.4.6).

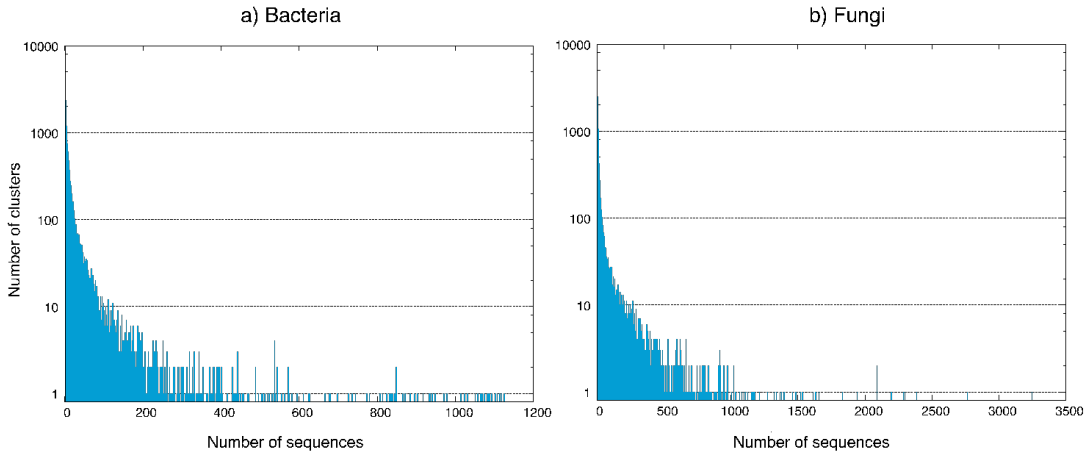


Figure 2.10: Distribution of cluster size for the full bacteria and fungi datasets with one representative and putative subsequence homology.

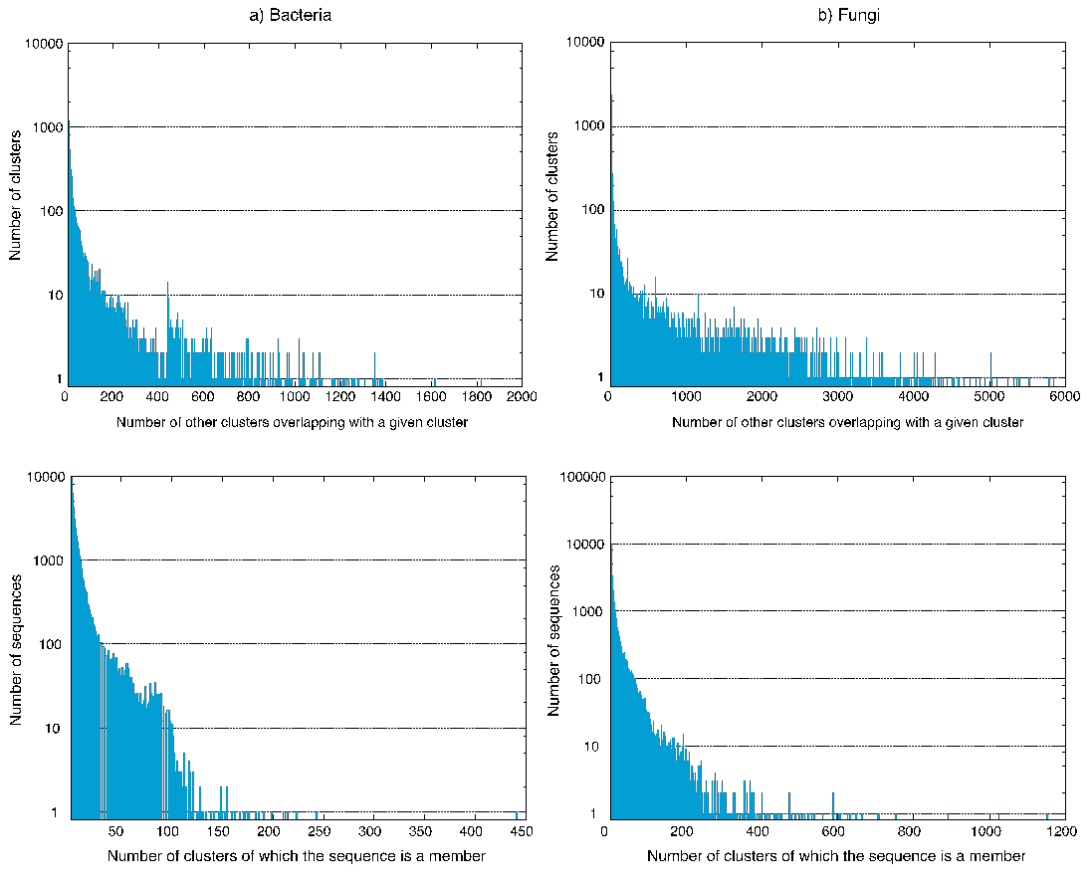


Figure 2.11: Histogram of the number of clusters overlapping with each cluster (top row) and of the number of clusters in which each putative sequence is involved (bottom row) for the full bacteria and fungi datasets with one representative and putative subsequence homology.

2.3.7 Slower but more accurate than *k*-mer methods

Although tested *k*-mer approaches are a lot faster than the new algorithm we propose (kClust (Hauser, Mayer and Söding, 2013) ~500-1,900x, UCLUST (Edgar, 2010) more than 6,000x; Fig. 2.12, Table A.7), they achieved low recall values when ran on the same bacteria and fungi datasets (kClust 6.62-13.52%, UCLUST 3.16-10.37%; Fig. 2.13, Tables A.8-A.9). Low recall can be partially attributed to the requirement of at least 30% sequence identity between query and cluster representative sequences in the corresponding clustering procedures.

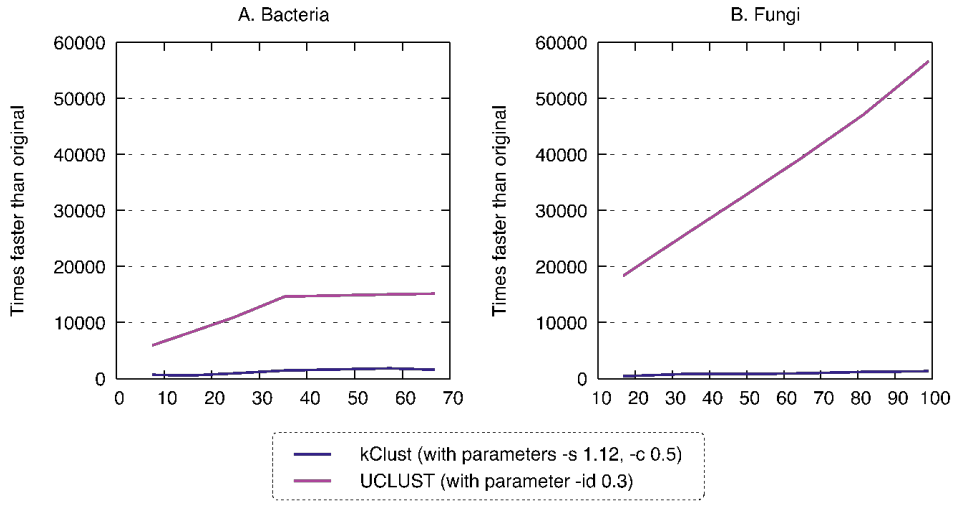


Figure 2.12: Runtime comparison of the new approach with kClust (Hauser, Mayer and Söding, 2013) and UCLUST (Edgar, 2010) algorithms.

kClust and UCLUST are several orders of magnitude faster than full all-against-all. Due to the low resolution, the plot shows outcomes of the kClust and UCLUST experiments which achieved higher recall. kClust with parameters `-s 1.12, -c 0.8` was faster than the one depicted here while UCLUST with parameters `-id 0.3, -target_cov 0.5, -maxaccepts 0, -maxrejects 0` was slower than the one on the figure. Runtimes of all kClust and UCLUST experiments can be found in Table A.7.

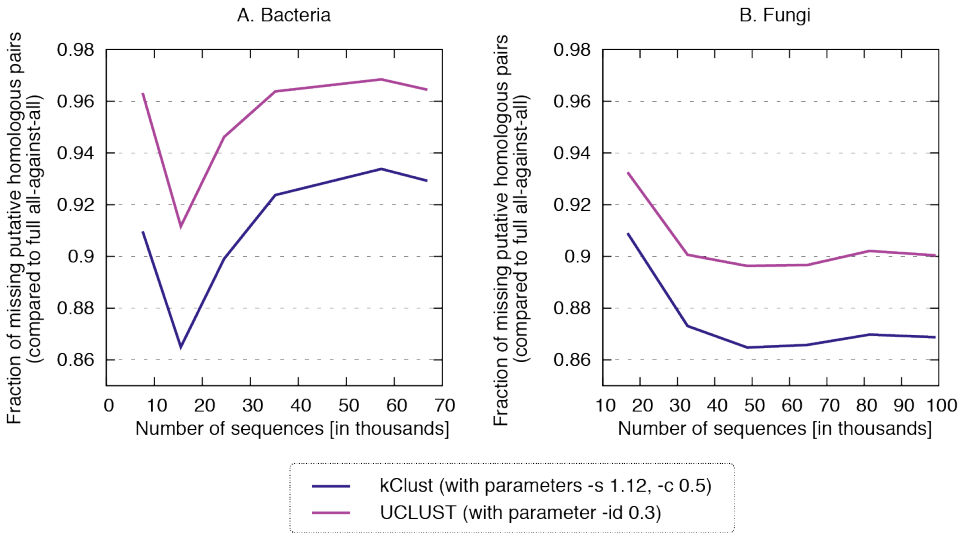


Figure 2.13: Fraction of the OMA putative homologous pairs which are not identified with kClust (Hauser, Mayer and Söding, 2013) and UCLUST (Edgar, 2010).

kClust and UCLUST only recover 3-14% of homologous pairs predicted by the full all-against-all procedure. Due to the low resolution, the plot shows outcomes of the kClust and UCLUST experiments which achieved higher recall. Their performance with other parameters can be found in Tables A.8 (for kClust) and A.9 (for UCLUST).

2.3.8 Code availability

Darwin (Gonnet *et al.*, 2000) implementation of our best variant (accounting for putative subsequence-level homology (defined in section 1.4.2) and using a single representative sequence) is available as part of the open source OMA standalone package (http://omabrowser.org/standalone).

2.4 Discussion and outlook

In the following sections we discuss scientific contribution of our work, explain the nature of its limitations, argue its potential beyond the current framework and provide ideas for further improvements.

2.4.1 Incorporating transitivity of homology into homology prediction and considering putative subsequence homology can substantially speed up homology inference

The all-against-all phase is at the basis of many orthology inference algorithms but it can be a bottleneck due to its quadratic time complexity in terms of number of putative sequences. This work suggests that incorporating transitivity of homology[34] can substantially speed up homology inference while maintaining sensitivity—provided that the inference method considers putative subsequence-level homology (defined in section 1.4.2).

If homology always held across all domains of a protein and could be perfectly inferred from present-day putative sequences, each putative sequence would belong to one and only one cluster of putative homologs. In practice however, these assumptions do not hold: partial homology (see section 1.4.2) can result in sequences that have homologous relationships across multiple clusters; some homologs have diverged too far to be inferred as such, thus resulting in cluster fragmentation. By allowing for putative sequences to belong to multiple clusters, our method is robust to these complications. However, we observed a larger number of clusters than anticipated which in combination with their overlapping nature makes them hard to interpret. An extensive analysis of the clusters could help understanding how and where the transitivity breaks.

[34] In our method, transitivity is incorporated only for the triples (sequence 1, sequence 2, cluster representative) where putative homology can be detected for pairs (sequence 1, representative) and (sequence 2, representative).

2.4.2 Substantial speedup with runtime complexity possibly subquadratic in the number of species

The speedup we observed with the new approach is substantial. But because the number of clusters increased roughly linearly with the number of proteomes in our datasets of up to 14 putative proteomes, the overall time complexity still grows quadratically in this range. However, as rarefaction curves show (e.g. Mira *et al.*, 2010), the growth in the number of clusters typically tapers off. Thus, it is possible that the asymptotic complexity of the new procedure is subquadratic in the number of species. This will need to be confirmed in future work.

2.4.3 Good ability to find evidence for homology in sequences coming from evolutionary distant gene pairs

The datasets used in this study are challenging, with estimated median distance of 146 PAM (1.46 substitution per site on average) in the bacteria dataset and 149 PAM in the fungi dataset (Fig. A.4). This implies that half of putative homologous pairs have less than 35% sequence identity (Dayhoff, Schwartz and Orcutt, 1978). At such high levels of divergence, *k*-mer based methods perform poorly compared to all-against-all dynamic programming alignment. Admittedly, to maximise recall, we could have tried even more permissible settings for kClust (Hauser, Mayer and Söding, 2013) and UCLUST (Edgar, 2010). kClust could have been applied requiring, e.g. 20% sequence identity as it reportedly demonstrated good performance at pairwise sequence identities of 20% and 30%. As for the UCLUST, its manual (Edgar, no date) states:

> UCLUST is effective at identities of ~50% and above for proteins and ~75% and above for nucleotides. At lower identities, this type of method is questionable because (i) alignment quality degrades and (ii) homology cannot be reliably determined from an alignment.

On the datasets considered in our study, our approach performed well. It is likely to perform even better on evolutionarily closer sets of taxa, such as vertebrate species or flowering plant species.

2.4.4 Trade-offs and applicability of the approach

In developing the algorithm presented in this chapter, we explored a number of alternatives, described section A.1. The process was more focused on maximising the recall (summary in Table A.2) although we considered and measured the runtime as well (Fig. A.1). While the new algorithm avoids performing unnecessary pairwise comparisons, i.e. comparisons of putative non-homologs, it also introduces clustering-related computations. However, the clustering step should ensure the overall reduction of computations and, importantly, allow for recovering >99% (ideally 100%) of putative homologs identified by the full all-against-all approach.

The backbone of the new approach is an attempt to incorporate transitivity of homology. Yet in practice, the homology inference method is not transitive. To overcome modelling limitations and improve the sensitivity of the method, we increased the number of cluster representatives, allowed assigning sequences to multiple clusters and considered putative subsequence-level homology, i.e. introduced coverage criteria—all at the cost of a longer runtime. The resulting clusters also contain putative non-homologous pairs and an all-against-all procedure within clusters is required to compute a final set of putative homologs. Compared to the established full all-against-all procedure which we intended to speed up, the new approach performs fewer pairwise comparisons but also infers fewer putative homologs.

The recall is directly affected by the choice of cluster representatives. In the current setting, representatives depend on the order in which sequences are processed—they are the founding sequences of clusters and they do not change as the cluster content changes. Furthermore, the number of representatives per cluster is fixed and the same for all clusters regardless of

the cluster content, i.e. the diversity of putative sequences within a cluster, and thus, regardless of the diversity of putative sequences that should be assigned to the cluster. Although it reduces the number of computations and it worked well on the tested datasets, such a naive approach is probably not optimal for many datasets.

There are also further problems with assigning putative sequences to the clusters. A sequence is aligned only to the cluster representative(s) and their pairwise score is compared to an arbitrarily chosen and fixed threshold (137.5). The scoring matrix—224 GCB (Gonnet, Cohen and Benner, 1992)—is optimised for comparing pairs of sequences being 224 PAM units apart. Pairwise alignments for sequences derived from closer genes might not be optimal but are anticipated to score above 181 in case of putative homologs (according to Roth, Gonnet and Dessimoz (2008)). On the other hand, the dataset might contain more distant putative homologs and they might be missed due to the scoring matrix. Furthermore, the score threshold is the same for all clusters although not all homologous genes and families evolve at the same evolutionary rate. By avoiding additional computations to determine optimal cluster thresholds some homologous relationships are missed due to the suboptimal cluster assignment. Similarly, the coverage criteria is also fixed and the same for all clusters causing missed relationships when a sequence does not found a cluster and does not become a representative of any cluster. In addition, information on sequence assignment is not used to perhaps merge clusters which could increase both recall and speedup nor it is used in any way to exploit sequence divergence within clusters towards achieving higher recall.

Despite all drawbacks, the new approach performed well on the considered datasets. To be specific, it achieved a ~2-5x shorter runtime at the cost of missing <0.4% putative homologs on selected bacteria and fungi datasets (section 2.3.1), and most notably 12.05x speedup while missing 0.06% putative homologs in a diverse dataset (section 2.3.2). Based on our experiments, the faster the variant, the lower the recall (Fig. 2.14).

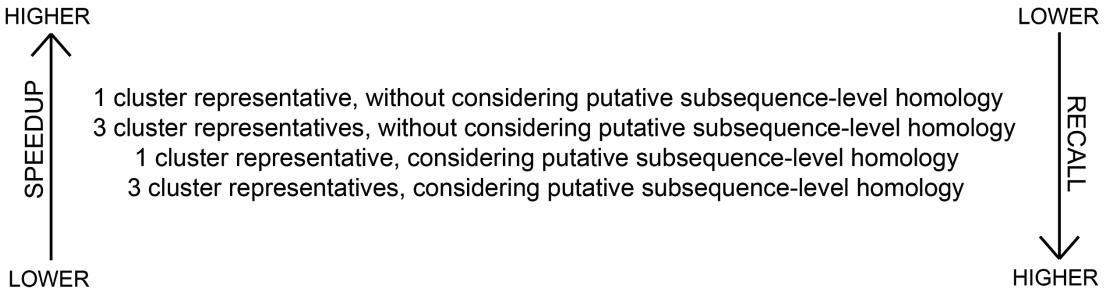


Figure 2.14: The trade-off between speedup and recall of tested algorithm variants.

Ordering based on the results presented in section 2.3.1.

One of the major drawbacks of the new algorithm compared with the standard all-against-all is the lack of parallelism. As such, it is not convenient for homology inference on larger datasets, particularly at a database level.

The proposed approach did not face any memory-related problems on rather small datasets used in the study (described in section 2.2.6). However, we are aware that memory will likely be an issue for a parallelised algorithm applied to a larger dataset. Thus, the way to keep only the minimum amount of information needed and preferably only temporarily in appropriate data structures should be addressed in the design of a parallel algorithm.

The approach can aid studies already in its current form. For example in projects where the gain in terms of runtime is of higher importance than higher sensitivity. It could be used as one of the methods when a union of predictions obtained by multiple homology inference methods is considered (potentially missed homologous pairs could still be inferred by another method). Speedup could be even higher on datasets comprising more similar putative proteomes such as in population genomics studies.

On the other hand, the method can miss meaningful homologous relationships and affect downstream analyses. A low percentage of missed putative homologs can translate into a large number in a dataset containing thousands of putative sequences. Missed putative homologs lead to missed putative orthologs, the lack of evidence in the process of structural and

functional genome and transcriptome annotation, smaller putative gene and protein families, less information in the multiple sequence alignments and phylogenetic trees, for example. Finally, missed homologous relationships can limit detection of fragmented gene models in genome annotation as will be discussed in section 3.4.3.

Conceptually, the algorithm is parallelisable, but implementing a scalable and robust parallelised approach would require a substantial amount of work. A parallel implementation would increase its applicability and further speed up homology inference. We believe it should take priority over other potential improvements in future work[35].

2.4.5 Potential beyond the current framework

Though the present method focuses on Smith-Waterman dynamic programming alignments (Smith and Waterman, 1981), a similar clustering approach would be possible with the faster but less sensitive BLAST (Altschul *et al.*, 1997). One complication with BLAST is that the sequence database needs to be re-indexed whenever a sequence is added. To mitigate this, one could add new representative sequences in batches, with the additional complication that sequences within each batch would also need to be aligned to one another.

A further advantage of the clustering approach is that it also works well in the context of "semi-curated" databases, such as in COGs (Tatusov *et al.*, 2003), PANTHER (Thomas *et al.*, 2003) or Pfam (Finn *et al.*, 2014). Indeed, it is conceptually straightforward to let curators optimise particular clusters of putative homologs by fine-tuning representative sequences, coverage and score thresholds on a cluster-by-cluster basis.

[35] The decision not to pursue a parallel implementation in the rest of the thesis is not inconsistent with this observation. It merely reflects prioritisation of a distinct (but complementary) line of investigation.

2.4.6 Further improvements

Although recall is high, above 99% in most of our runs, even a small fraction translates into many missed putative homologs when working with huge datasets or databases. Hence, further improvement is desirable. Several ideas could be explored.

One of the most obvious potential modifications is tuning the alignment score threshold for cluster assignment. In our preliminary studies we tried out only two thresholds (181 as in the full all-against-all algorithm and ¾*181=135.75—an arbitrarily chosen fraction of 181). The smaller the threshold, the bigger the clusters and possibly the fewer missed putative homologous pairs. Yet, some previously identified pairs might get missed as some sequences would not found clusters and hence, would not be compared to the sequences processed after them. One could also consider adapting the threshold to the particular family as different families evolve at different evolutionary rates. For example, an initial threshold could be set to 135.75 as it is now. As the sequences are processed, pairwise alignment scores between cluster representatives and query sequences could be memorised, even for pairs which do not end up in the same cluster. If a cluster tends to contain sequences with scores closer to the threshold, the threshold could be lowered. Furthermore, previously compared but not included sequences that satisfy the new criteria could be added to the cluster. This would introduce more bookkeeping and require more memory, the resulting clusters would be larger and require longer runtime in the all-against-all step, but the procedure could yield higher recall. However, it remains unclear how scalable such an approach would be.

The alignment score is calculated using 224 GCB matrix (Gonnet, Cohen and Benner, 1992) which is expected to serve well when comparing sequences being up to 224 PAM units apart, as elaborated in section 2.4.4. To be able to find indications for homology among even more distant sequences, other matrices could be used instead. However, the choice of scoring matrix should take into account the whole dataset and not just pairs

having a certain estimated evolutionary distance. Hence, rather than picking a matrix which could help detecting even more distant pairs (e.g. 400 GCB matrix), it would be better to estimate the optimal GCB matrix for a particular dataset, as did Roth, Gonnet and Dessimoz (2008) for a general case.

When considering subsequence-level homology, we tolerate up to 20 AA of a putative sequence not being covered by a cluster representative. Lowering the tolerance might help to increase the recall, yet at the cost of creating more clusters. Similarly as the threshold for pairwise alignment score, the coverage tolerance could also change depending on a putative family.

Another promising avenue could be improving the choice of cluster representatives. Indeed, the current strategy of selecting the first putative sequence (or the first three putative sequences) added to the cluster is likely to be suboptimal in most instances. Instead, a better choice of representative would be to try to select a putative sequence with estimated minimal average distance to all cluster members (a “centroid” sequence). Calculating a consensus sequence of all sequences in a cluster and setting it as a representative could increase the size of clusters but we anticipate that the algorithm would show similar behaviour to the variant with the longest sequence as a cluster representative (section A.1.1)—as the consensus sequence becomes longer, the cluster content will likely diverge more from the founding sequence(s). As opposed to the longest sequence, we did not investigate the possibility of taking the shortest sequence as a cluster representative. Due to the coverage criteria, it could lead to the creation of a large number of clusters. Although the assignment of representatives would be dynamic, the risk of divergence from the founding sequences might be lower due to the short length of representatives (given the same pairwise alignment score threshold as it is now).

Another idea would be to vary the number of representatives depending on the particular needs of each cluster. For example, the number of representatives could be increased if the distribution of pairwise alignment scores between representatives and sequences in the corresponding cluster

is skewed or multimodal. A less computationally demanding option could be to add a new member as a representative if its similarity score with the existing representative(s) is below a certain threshold. Also, instead of creating new clusters when all representatives exceed the coverage tolerance with a query sequence, a new sequence could be added to the set of cluster representatives. Adding more representatives instead of creating new clusters would lead to bigger clusters and lower number of clusters, hopefully increasing the recall of the method. Since it would require more all-against-all computations within clusters, it would be good to get a good grasp on the trade-off between sensitivity and runtime for a chosen algorithm modification prior to its applications to large datasets.

The choice of cluster representatives depends on the ordering of processed putative proteomes and their sequences. In the current version, input putative proteomes are sorted by the number of putative protein sequences and processed starting with the largest proteome first (in terms of sequence numbers)—a decision made based on application on a single small dataset (section A.1). This could be further explored as well as the effect of ordering protein sequences within putative proteomes. For example, protein sequences within putative proteomes could be sorted by their length and processed in descending order with the hope that this would yield longer cluster representatives and a smaller number of clusters. Furthermore, all sequences in the dataset could be sorted according to their length and processed starting with the longest regardless of the putative proteome they belong to. Putative proteomes could also be processed in the order determined by estimated evolutionary distances between species. For example, we could start with the closest two in the dataset and continue by moving to the next closest species until all putative proteomes are processed. Moreover, we could also increase the number of cluster representatives as we move to the more distant species. This could yield higher recall at the cost of moderately higher runtime. One more idea is to sort putative proteomes by their quality in terms of (estimated) coverage and completeness, and process starting with the best one. Again, the number of cluster representatives could be increased as the data quality decreases.

This approach could yield more accurate homology inference among higher quality putative proteomes as it would not be challenged by the lower quality data. It could also ease homology inference for the low-quality sequences given the choice of representatives.

Crucially, an empirical analysis of true homologs and their pairwise alignment scores could provide insights for better clustering including the choice of cluster representatives, thresholds for cluster assignment and coverage requirements. However, a dataset containing only experimentally validated protein families (e.g. Higgins (1992), Nakanishi (1992), Jacoby *et al.* (2006)) might be too small to extrapolate conclusions across the tree of life. A more comprehensive investigation could include manually curated datasets (e.g. Byrne and Wolfe (2005), Boeckmann *et al.* (2011), Trachana *et al.* (2011), Gray *et al.* (2016)) but the generality of conclusions is also not clear. Another way to perhaps gain meaningful insights could be through a simulation study. Proteome sequences could be simulated using, for example, Artificial Life Framework (Dalquen *et al.*, 2012) which also outputs protein families. That would enable investigating a wider variety of evolutionary ranges and sequence lengths, but artificially ones. Based on these analyses, the users could be provided with parameter recommendations. Yet, the applicability and scalability of these suggestions remain uncertain.

Perhaps the problem of selecting cluster representatives could be easier if each cluster represented a single feature, such as a protein domain. As in the current algorithm, each sequence could belong to multiple clusters at the cost of creating a large number of clusters but with the potential advantage of higher sensitivity. Furthermore, the runtime could be improved with a strategy on merging clusters (discussed later in the section). Since a shared domain does not necessarily imply homology at a gene level, an all-against-all step within clusters would be of particular importance. Putative homologs could also be assessed using approaches that attempt to distinguish putative homologs from sequences that just share a putative domain (e.g. see Song *et al.* (2008)). Other examples of features include GC-content, sequence

length and, generally speaking, any statistical sequence features or their combination which could characterise a putative family (Brendel *et al.*, 1992).

The procedure could also merge clusters to attempt to recover some of the missed putative homologs. Candidates for merging could be those having a high proportion of members in common. Merging could also be considered based on high pairwise alignment score or protein domains shared across representatives of different clusters. Instead of comparing cluster representatives, consensus sequences for the clusters could be examined as well.

We also see potential to further improve the speed of the new approach.

First, merging clusters can lead to a speed improvement because the cost of assigning putative sequences to clusters grows linearly in the number of clusters. This needs to be done carefully, because excessive merging—the merging of clusters containing a substantial number of putative non-homologous pairs—can reduce the efficiency of the within-cluster all-against-all, whose time complexity grows quadratically in the number of sequences.

Second, it may be possible to optimise the assignment of putative sequences to clusters by identifying clusters that are so different to one another that they are practically mutually exclusive and thus inclusion into one implies exclusion from the other. An empirical way of establishing such mutual exclusivity would be to keep track of the number or proportion of sequences belonging to both clusters. After processing a certain large number of putative sequences, mutual exclusivity could be deemed, yet at the risk of having unprocessed putative protein sequences of fusion genes and more generally, putative protein sequences belonging to both clusters given the thresholds.

Third, a bottom-up tree-guided modification of the clustering approach could also be considered. Leaves of a reconstructed species tree could contain putative proteomes, each processed with the current clustering method. A

parental node of two leaves could hold all their clusters, some of them merged. For merging clusters in an internal node, cluster representatives from the left child could be compared to the cluster representatives from the right child, and merged based on their pairwise alignment score. More than two clusters could be merged into a single one. The resulting cluster could keep all old cluster representatives as its own representatives to increase the sensitivity. Given that once at the level of internal nodes, only cluster representatives are mutually compared, the amount of computations should be feasible despite the increased number of cluster representatives. Using the bottom-up strategy, internal nodes should be processed until reaching the root which will contain a final set of clusters. This is just an idea that has yet to be tested to get a better overview of its behaviour in terms of the runtime and, importantly, recall.

Finally, the current approach could be parallelised. One way of parallelising the assignment of putative sequences to clusters would be to use a Publisher-Subscriber model (Eugster *et al.*, 2003): a “master” process would start the analysis of a new putative sequence by distributing it to a set of “workers”, each responsible to compare the sequence to a subset of all existing clusters. Each worker would thus align the new sequence to its designated subset of clusters and report back significant matches and their associate sequence ranges (subsequence-level coverage). Once the master process has received this information from all workers, it would ensure that the new sequence is fully covered by matches to existing clusters, and else it would generate a new cluster with that sequence as a representative. As for the within-cluster all-against-all comparisons, they could be straightforwardly parallelised thanks to the lack of dependency among all pairs. However, it is not clear how scalable the model will be in practice because of the communication between the master process and the set of workers. Furthermore, the algorithm might face memory issues if the bookkeeping is poorly optimised in terms of the amount of information memorised, its timespan and the choice of data structures.

2.4.7 Potential improvements through profile and profile HMM cluster representations

As an alternative to representative sequences, the clusters could be represented by profiles or profile hidden Markov models (HMMs) (concepts described in section 1.4.3). Accounting for information from multiple sequences and being concerned with position-specific modeling, such approaches are likely to be more sensitive than pairwise alignment with a single representative sequence which is of particular interest for detecting more distant homologs having sequence identity between 20 and 35% (Pearson, 2013; Chen *et al.*, 2016). However, the lack of information in the data, information from wrongly assigned cluster members and compromises made in the modelling procedure affect profiles and profile HMMs. Thus, they can have an adverse effect on the remaining clustering assignment. That could lead to false positive assignments of putative sequences to the clusters as well as to the false negative assignments. For those reasons, we would still allow sequence assignment to multiple clusters and perform the all-against-all comparisons within clusters.

Profiles representing clusters could be built along the lines of the PSI-BLAST approach (Altschul *et al.*, 1997), for example. The first two or several cluster members could be determined as in the current algorithm and used to calculate corresponding cluster position-specific score matrix (PSSM). When the next putative sequence is inspected for potential cluster membership, the score would be computed using the PSSM instead of pairwise alignment to the cluster representatives using position-independent 224 GCB scoring matrix (Gonnet, Cohen and Benner, 1992). With each new cluster member, the PSSM could be recalculated to account for information from all cluster members. This would require a lot of additional computations, so perhaps under the assumption that after a while the newly added sequences would contribute less and less information, the PSSM could be recalculated after several new putative sequences are added to the cluster. Nevertheless, the risk of wrongly assigning or not assigning sequences to the clusters would be higher than when recalculating the PSSM with each new cluster member. A

decision on updating the PSSM could also be made based on the score of the newly added sequence with the cluster profile, i.e. updating the PSSM if the score is below a certain threshold. With PSSM recalculations, the approach could have longer runtime than the current one but with potentially higher recall (could be anticipated based on, e.g. Altschul *et al.* (1997), Pearson (2013)). However, the additional runtime is not the only issue here; extracting a part of the cluster multiple sequence alignment that will be used for calculating a PSSM, assigning weights to the involved truncated sequences (the more close relatives in the set, the lower the weight of a putative sequence), building a context-specific profile (taking into account neighbouring residues of a position) or not, treatment of gaps (fixed or position-specific gap costs), choosing a method for calculating matrices and setting score thresholds for cluster assignments—all bring in more optimisation problems some of which have to be tackled on a per cluster basis. Furthermore, inclusion of non-homologs into the putative family directly affects the PSSM and further cluster assignments (Pearson, 2013). The runtime could be reduced by, e.g. starting from the PSSMs available at the NCBI CDD Database (Marchler-Bauer *et al.*, 2016). Since profiles can also represent putative protein domains, they could facilitate grouping together putative sequences sharing a particular putative domain, i.e. creating clusters representing a single feature. PSSMs for conserved protein domain detection could also be downloaded from the NCBI CDD Database. Depending on the known and represented domains, domain-based clustering could further increase the recall, as already mentioned in section 2.4.6, but that would also require more comparisons to the cluster profiles in the clustering step and more comparisons in the all-against-all step.

Analogous but a more sensitive approach than generating profiles could be to represent each cluster by a profile hidden Markov model (HMM) like in, for example, HMMER3 (Eddy, 2011), SAM-T98 (Karplus, Barrett and Hughey, 1998) or HHblits (Remmert *et al.*, 2011). A profile HMM could be created from a single cluster founding sequence or after more sequences have been clustered together using the existing method. If the

$log \frac{P(sequence\ was\ generated\ by\ the\ profile\ HMM)}{P(sequence\ was\ generated\ by\ a\ random\ profile\ HMM)}$ is above a certain threshold, the putative sequence would be assigned to the corresponding cluster. Alternatively, a query sequence could also be represented by a profile HMM and compared to the cluster profile HMMs. If the distance between the HMMs (e.g. Kullback-Leibler divergence (Falkhausen, Reininger and Wolf, 1995)) is below a chosen threshold, the sequence could be added to the corresponding cluster. Again, a profile HMM could be updated with each new putative sequence added to the cluster or periodically at the risk of missing some putative homologs. As with cluster profiles above, profile HMM representation comes with its advantages and disadvantages. It could yield higher recall (Chen *et al.*, 2016), and even higher recall is anticipated for HMM-HMM comparisons (Remmert *et al.*, 2011), yet at the cost of more computations, i.e. longer runtime. Each profile HMM would require model construction, parameter estimation on the training data and setting a log-odds (or distance) threshold for cluster inclusion of new members (Durbin *et al.*, 1998). To speed up the procedure, precomputed profile HMMs could be downloaded from established sources, such as those of InterPro Consortium (EMBL-EBI, 2017), and used as a starting point for clustering the dataset under investigation. Profile HMMs can also represent protein domains and provide a backbone for a domain-based clustering. Again, relying on already available profile HMMs (e.g. from InterPro Consortium) would speed up the clustering procedure and yield recall dependent on the represented domains.

Unlike the current algorithm, a conventional clustering procedures with profile and profile hidden Markov model (HMM) cluster representations would not consider putative subsequence homology (the term explained in section 1.4.2). Consider multi-domain proteins: if profile HMMs are constructed over the full putative protein length, such as in the PANTHER database (Mi *et al.*, 2019), the model might not align well with putative sequences that only share a subset of the domains. As a result, some of these pairs, which might still be classified as putative homologs in a pairwise context (given typical alignment length tolerance parameters) might not fit to the same profile HMM model. Alternatively, it might be possible to build profile HMM models at the level of

domains, such as in Pfam (Finn *et al.*, 2014), but going from domain-level homology inference to gene-level homology inference would not be straightforward, and might result in quite different homolog predictions than the conventional all-against-all alignment procedure which we have used as baseline in this chapter.

Overall, replacing cluster representatives with cluster profiles or profile hidden Markov models (HMMs) might help detecting a higher number of putative homologs but at the cost of solving more optimisation problems and perhaps even longer runtime of the approach. An algorithm modification with profile HMMs might yield higher recall than a modification with profiles while the latter one might be faster, as mentioned for already existing tools in section 1.4.3. However, both recall and runtime will depend on a particular algorithm, its parameters and additional resources (training datasets, precomputed profiles or profile HMMs), if used. For example, all putative sequences in a dataset could be compared against publically available profiles or profile HMMs without updating the models. Some sequences would get assigned to clusters and some would not. Then only for those unclustered sequences, profiles or profile HMMs could be constructed. Although it might be less computationally intensive, such an approach could still be quite challenging in terms of model building.

Chapter 3: Phylogenetic heuristics to identify fragments of the same gene model in low-quality putative genomes, with application to the putative wheat genome

3.1 Introduction

As already elaborated in Chapter 1, one problem in low-quality genome assemblies is that fragments derived from the same gene can be annotated as distinct entries in genome databases and affect later analyses and applications. However, it is possible to use putative homologous proteins conserved in other species to detect fragments that are likely to be part of the same gene model, i.e. sequence fragments derived from the same gene.

To address this problem, we present two complementary heuristic phylogenetic methods to identify non-overlapping or slightly overlapping fragments of the same gene model that exploit inferred evolutionary relationships across putative gene families. The first one exploits SH-like branch support (Shimodaira and Hasegawa, 1999; Guindon *et al.*, 2010) and the second one relies on a likelihood ratio value (Edwards, 1972). We evaluate their performance on an artificially fragmented annotation of the bread wheat chromosome 3B reference sequence assembly (as of 2014-2018). We also compare the two methods and ECOMB, a meta-approach combining the two methods with ESPRIT (Dessimoz *et al.*, 2011), to the Ensembl Compara pipeline (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020) and ESPRIT. Finally, we apply new phylogeny-based heuristic methods to the early, highly fragmented, draft release of the entire putative bread wheat genome and identify 1,221 pairs of split gene models.

3.2 Methods

We first introduce our heuristic phylogenetic tests of split gene models providing the fine implementations details, then proceed to describe the datasets analysed and the evaluation methods.

3.2.1 Reasoning behind the tests

Given a genome assembly with a large number of annotated contigs, the task we face is to figure out which annotated gene sequences are actually derived from the same gene, but have been misannotated to separate genes due to annotation mistakes or the failure of the assembler to concatenate collinear contigs. Consider therefore two non-overlapping fragments of the same gene model. If we perform a multiple sequence alignment of the two fragments together with full-length putative homologous sequences from other species, we can expect that the two fragments align to different regions of the multiple sequence alignment. If we infer a phylogenetic tree based on the alignment, these fragments will almost never be inferred as sister leaves (adjacent tips; also known as “cherries”) although both fragments started diverging from their homologous counterparts at exactly the same moment, hence in the biologically true tree they should branch off at exactly the same place. The reason for getting different estimated branching off points for the fragments is that since they have no character in common, they cannot be directly compared to each other, only to the rest of the putative sequences. Thus, there is no phylogenetic information available to infer the relationship between them, only to the rest of the tree. The location of the split between the two sequences is therefore undetermined. Furthermore, recall that evolution is modelled as a stochastic process on a tree, with each column in the alignment being a realisation of the process. The two fragments will almost certainly consist of different realisations. Therefore, in the maximum likelihood estimate of the tree, the two fragments’ terminal branches will almost never attach to the exact same place on the tree. Under a model of evolution where sites are independent and identically distributed, such as

those that are commonly used in phylogenetic inference (e.g. LG+I+Gamma[36]), if the two fragments originate from the same gene, we can expect that both estimated branching off points will be within the estimation error, so insignificantly distant from the true branch point, and also insignificantly distant from one another.

Hence, for two non-overlapping fragments of the same gene model, we can expect that they: i) align to different regions of the multiple sequence alignment, and ii) generally sit close to one another separated by insignificant branches in a gene tree inferred from the fragments and their putative homologs.

3.2.2 Test #1: Collapsing insignificant branches

Consider a maximum likelihood tree for a set of putative homologous gene sequences containing fragments of the same gene sequence. As explained thoroughly in section 3.2.1, the fragments' terminal branches are likely attached close to one another on the tree but very rarely at the exact same place. Yet, if the tree is built under a model of evolution with sites being independent and identically distributed, the attachment points should be separated by insignificant internal branches. Hence, collapsing insignificant branches should result in fragments becoming sister leaves.

Tree branch support measures are commonly used to gauge the reliability of a branch. Thus, we propose a heuristic test that, for a given threshold, collapses all branches below that threshold and infers as fragments of the same gene model all candidates that are sister leaves (example in Fig. 3.1). Note, however, that paralogous genes can also code for sufficiently similar sequences that can end up being placed on sister leaves in the collapsed tree. Hence, being on sister leaves can only indicate that fragments could be

[36] LG amino acid substitution matrix (Le and Gascuel, 2008); invariant + gamma model of rate heterogeneity (Yang, 1994; Gu, Fu and Li, 1995)

derived from the same gene but does not provide evidence for it being the true scenario.

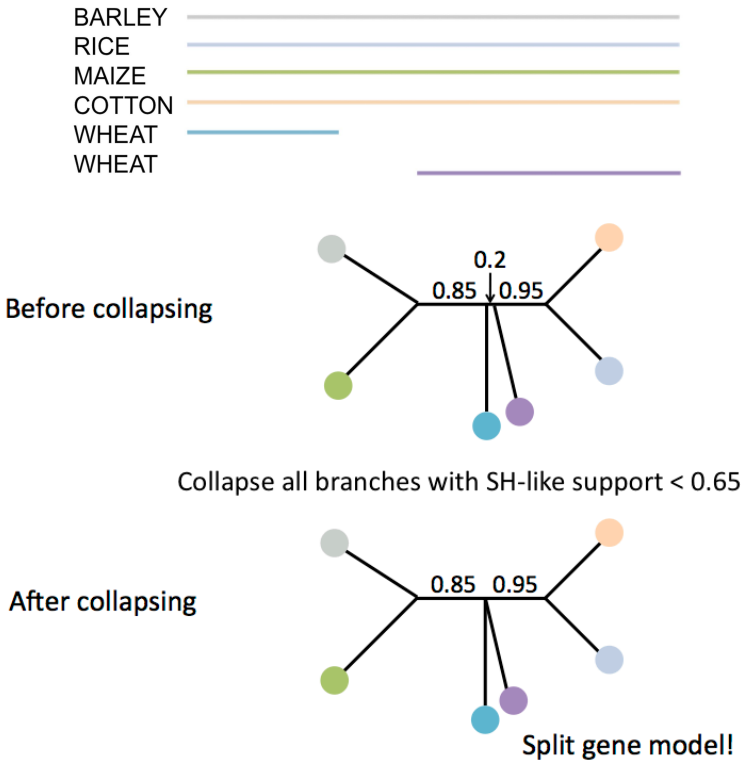


Figure 3.1: An example of application of the collapsing approach with collapsing threshold of 0.65.

Given an MSA (depicted at the top of the figure), we would like to test if blue and purple wheat gene models are fragments of a single gene model. After building a gene tree for the corresponding putative gene family and collapsing tree branches with SH-like support lower than the chosen threshold, leaves corresponding to the fragments under investigation become sister leaves. Hence, according to the collapsing heuristic method, a split gene model is inferred.

3.2.3 Test #2: Likelihood ratio heuristic (LRH)

The second test we propose to infer fragments derived from the same gene is a likelihood ratio heuristic (LRH) defined as follows. Our null hypothesis (labelled "*s*" for *split)* is that fragments are parts of a single (longer) gene model, and can thus be concatenated and annotated as such. The

alternative hypothesis (called "*p*" for *paralogs*) is that the two non-overlapping sequences are coded by paralogous genes.

H_s: *n-1* gene models (split gene model)
H_p: *n* gene models (gene models on sequences coming from paralogous genes)
The likelihood ratio value is defined as $T = 2ln\frac{L(Hp)}{L(H_S)}$, where *L()* denotes the maximum estimator under each hypothesis (Fig. 3.2).

Likelihood ratio heuristic:

select H_s if $T < c$;

otherwise,

select H_p,

where the critical value *c* is such that $P(\text{reject } H_s | H_s \text{ is true}) = \alpha$ for a chosen test significance level α.

Equivalently, the test can be performed using the *p*-value approach:

select H_p if $p \leq \alpha$;

otherwise,

select H_s,

for the same α as above in the critical value approach.

Hence, if the gene models under examination are compatible with the hypothesis that they are derived from the same gene (H_s), we select the hypothesis. Otherwise, we select the alternative hypothesis (H_p) that they are derived from paralogous genes. Selecting the H_s does not mean it is true, nor does it mean that the H_p is not true, since putative paralogs can be arbitrarily close and sometimes they are even identical at the protein sequence level. Furthermore, when selecting the H_s, we do not investigate the distribution of the likelihood ratio value under the H_p—we do not select the H_p only based on the fact that the sequences tested are compatible with the H_s under the H_s.

Thus, the test is a heuristic which only allows an indication that fragments might be from a single gene to be found, but cannot provide evidence for the truth.

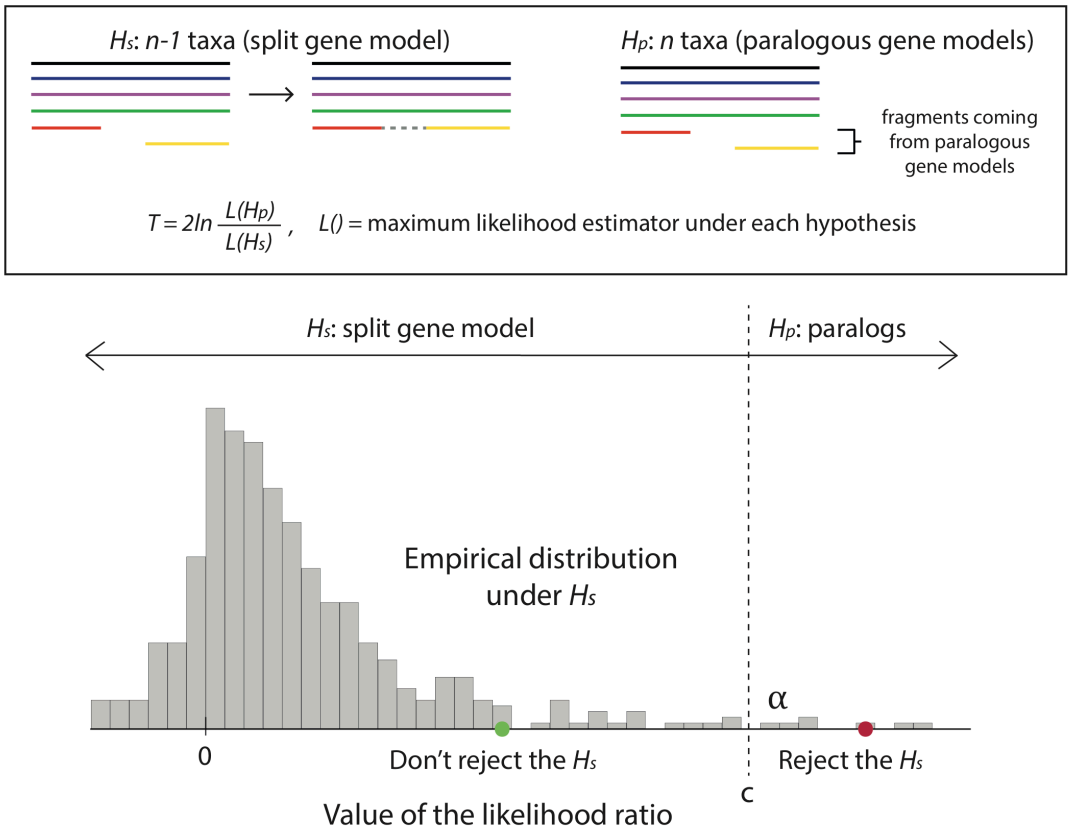


Figure 3.2: Conceptual overview of the likelihood ratio heuristic. The null hypothesis is that the two putative sequences come from the same gene and thus, a single gene model should encompass them (H_s) while the alternative hypothesis is that the two putative sequences come from paralogous genes and thus, associated gene models should remain distinct (H_p). This setup is motivated by the fact that the split gene model hypothesis has fewer parameters. Unlike the statistical hypothesis testing, failure to reject the null hypothesis leads here to a prediction, but furthermore the rejection of the null hypothesis leads to inference of a relationship among gene models under investigation.

Given the test definition, a practical question emerges—how to calculate the *p*-value? In a typical setting of the likelihood ratio test, the null model is a special case of the alternative model and the test statistic (analog here is the ratio of likelihoods *T*) is chi-square distributed (Wilks, 1938). Since our models are not nested, the distribution of the ratio of likelihoods under the assumption that H_s is true is unknown. This problem can generally be bypassed by estimating the empirical distribution under the null hypothesis using bootstrapping (Efron and Tibshirani, 1993; Goldman, 1993). Hence, for a particular sample, we could:

1. Compute the ratio of likelihoods; let's denote it by T_0
2. Since we have no prior knowledge on the distribution of the ratio of likelihoods under the null hypothesis, we could estimate the distribution using non-parametric bootstrapping. First, from the multiple sequence alignment used under the H_s we could generate *n* artificial alignments of the same length, i.e. *n* bootstrap samples by sampling columns with replacement. Second, we could create alignments to be used under the H_p by splitting a target *full-length* gene model (i.e. the one made up of two candidate fragments) at the same position as in the original alignment. Finally, we could compute the ratio of likelihoods for each of the *n* samples; let's denote them by $T_1^*, T_2^*, \ldots, T_n^*$.

   If the sampling is correct, the distribution of T_i^*, $i = 1, 2, \ldots, n$ will converge to the true distribution of the ratio of likelihoods when $n \to \infty$. Hence, if repeated many times, the distribution of the bootstrap sample ratio of likelihood values will approximate the distribution of the unknown ratio of likelihoods.
3. Compute bootstrap *p*-value as the proportion of samples with likelihood equal or above that of the input data: $p_B = \frac{\{\#\ of\ T_i^* \geq T_0\} + 1}{n + 1}$.

The problem with the above described procedure again lies in its assumption that the two putative sequences under examination are derived from the same gene. If the sequences are derived from two paralogous genes, concatenating them for the purpose of computing a tree under the H_s

produces a sequence and a tree that do not exist in reality. Furthermore, if the sequences are from different genes, the bootstrapped candidate fragments will also represent mixtures of the two locations for the paralogous genes on the phylogenetic tree and will not correspond to a set of positions corresponding to a single location for a single real gene. Again, the phylogenetic tree construction will be constructing a tree for a situation that does not exist. No matter the number of bootstrap samples, the distribution of the ratio of likelihoods for the null hypothesis will not be approximated.

A similar scenario would happen even if the hypotheses were reversed, i.e. if the working assumption was that the fragments had been derived from paralogous genes. Again, concatenating fragments derived from paralogs into a single putative sequence (H_s) yields a sequence that might not exist while bootstrap sampling produces fragments which are mixtures of the two locations on the tree. The latter could be bypassed by breaking the alignment into two parts corresponding to the regions spanned by each fragment and sampling from each side separately. Yet, what if the fragments indeed capture different parts of the same gene?

Nonetheless, we show below that when implementing the heuristic as described above—using the defined test criteria, assuming that candidates are parts of the same gene model, following the outlined steps 1.-3. for calculating *p*-value—the approach is able to detect fragments coded by the same gene.

3.2.4 Implementation of the tests

As input candidate pairs, we identified, among all the putative protein sequences of gene models in a putative target genome, those that belonged to the same putative protein family—either established by Ensembl Compara (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020) or defined as deepest (top-level) hierarchical orthologous groups as inferred by OMA (Altenhoff *et al.*, 2013). We further required fragments to be non-overlapping,

or overlapping with less than 10% residues of both fragments being aligned in the same alignment column, using Mafft v7.164b (Katoh and Standley, 2013). In other words, we required that $a_{12} < 0.1 * l_1 \; AND \; a_{12} < 0.1 * l_2$, where l_1 and l_2 are the number of residues in the two fragments, and a_{12} is the number of these residues that are aligned in the same column. Thus, for each inferred protein family, we aligned the sequences, enumerated all possible pairs of sequences belonging to the putative target proteome, and retained as candidate pairs those that satisfy the aforementioned overlap requirement.

The collapsing approach relies on the branch supports calculated for a reconstructed phylogenetic tree. Tree building tools used in this project calculated local support values with the Shimodaira-Hasegawa test (Shimodaira and Hasegawa, 1999) as described in Guindon *et al.* (2010). Like Guindon *et al.* we also refer to them as *SH-like branch supports*.

Our likelihood ratio heuristic requires computing maximum likelihood estimates, i.e. finding an optimal tree under both H_s and H_p. Under the H_s hypothesis, fragments are part of the same gene model. Hence, the input putative gene family has one sequence fewer than the input family for the H_p model. Consequently, the number of terminal branches (leaves) in resulting trees differs by one.

To get an input dataset in which putative protein sequences corresponding to candidate gene model fragments are part of the same putative protein sequence, we aligned a putative protein family and then replaced the two fragments with a newly created sequence containing residues from both fragments and gaps at the remaining positions. In case of non-overlapping sequences, this was straightforward. If sequences overlapped, we first determined the *middle* of the overlap and edited the sequences as follows. In the sequence on the left, we kept all positions the same up to the *middle* and replaced the remaining residues with X's. Similarly, in the sequence on the right, we edited the beginning of the sequence by replacing all residues up to the *middle* with X's and kept all remaining residues as they were. This way

we got two fragments with non-overlapping known residues in the alignment. They were used as such in both tests (Fig. 3.3).

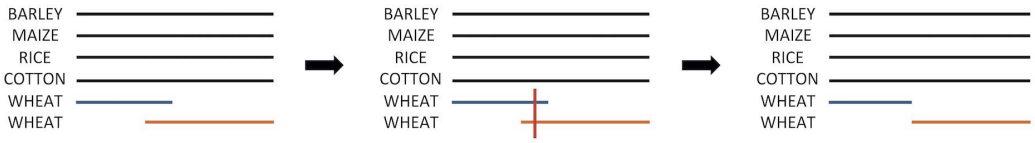


Figure 3.3: Dealing with slightly overlapping candidate sequences. We first determine the middle point of the overlapping region in the multiple sequence alignment. Then we trim the ends of sequences following or preceding the middle point in order to get non-overlapping input sequences.

To correct for at least some cases when a tree-building method gives a suboptimal tree, which may result in the estimated $T_0 < 0$ (impossible in theory; see section B.1), we performed two tree searches under the H_p model; a tree search without providing an input topology, and a tree search with an input topology. For the input topology, we modified the best tree under the H_s model. We bifurcated a terminal branch of the putative protein sequence corresponding to the candidate gene model, set new branches' lengths to 0 and the support of a branch leading to fragments' parental node to 0.5 (Fig. 3.4). Having performed both tree searches, we proceeded with the output tree with higher likelihood. Note that this tree search procedure does not guarantee to find true maximum likelihood tree under the H_p model; it only expands the search for it[37]. Also, since the goal of our method is to infer split gene models, in which case the maximum likelihood under H_p is only moderately higher than under the H_s, putting more effort into maximising the likelihood under the H_p is conservative (if not "fair").

[37] We could do more to get closer to the true maximum likelihood tree under both models by using more exhaustive optimal tree search (e.g. by changing settings of a chosen tree building tool, changing random seed, trying various input topologies, using different tree building tools, etc.). For the time being, our method sets a baseline.

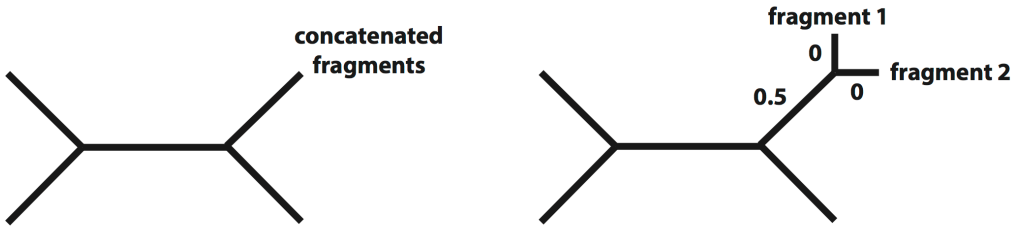


Figure 3.4: Input topology for the likelihood ratio heuristic.
Left: Maximum likelihood tree under the H_s. Right: Modified tree to be used as an input tree in the H_p model.

Throughout this project we set the number of bootstrap samples (n) to 100 unless otherwise stated.

3.2.5 Resolving multiple predictions and predictions with gaps

Some gene models might be involved in multiple predictions, i.e. in more than one pair of fragments of a split gene model (explained at the protein-sequence level in Fig. 3.5). If all these multiple predictions span different parts of the target gene model, we conclude that the gene model is split in more than two pieces and consider these predictions as unambiguous (Fig. 3.5a and 3.5d). If, by contrast, more than one prediction spans over a common part of the model (which might be the case if fragments are derived from very closely related paralogs, or if alternative splicing variants of the same gene are erroneously annotated as separate genes), we report the predictions as ambiguous (Fig. 3.5b-c).

When taken as a union, fragments involved in prediction(s) of a target gene model do not necessarily span the whole reference gene model length (Fig. 3.5d). This could be due to insertions in corresponding reference genes, deletions in the target gene or missing data. If unambiguous, we accept such predictions without investigating further reasons for gaps.

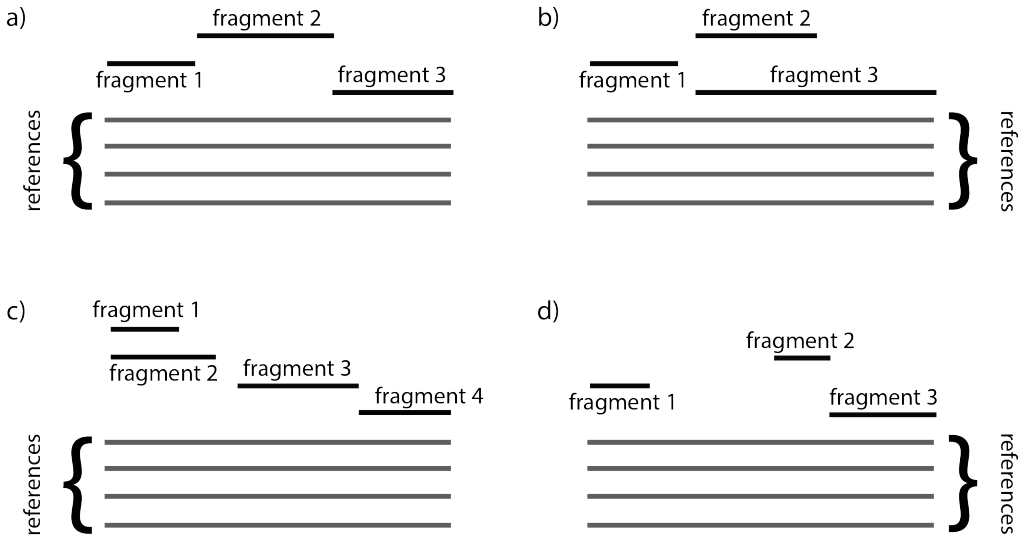


Figure 3.5: Resolving multiple predictions and gapped predictions. All four panels depict multiple sequence alignment (MSA) of putative protein sequences corresponding to the fragmented gene models involved in predictions (black lines) and putative proteins of reference gene models (grey lines). a) Unambiguous predictions. Predictions (fragment 1, fragment 2), (fragment 2, fragment 3) and (fragment 1, fragment 3) are unambiguous because fragment 1, fragment 2 and fragment 3 span different parts of reference sequences. b) Ambiguous predictions. Predictions (fragment 1, fragment 2) and (fragment 1, fragment 3) are ambiguous because fragment 2 and fragment 3 span the same part of reference sequences. c) An example where we classify a prediction as ambiguous due to ambiguities in a different part of the MSA. Let's assume that the following pairs are predicted: (fragment 1, fragment 3), (fragment 2, fragment 3) and (fragment 3, fragment 4). Since fragments 1 and 2 span the same part of the alignment, we classify all three predictions as ambiguous although there is no ambiguity about prediction (fragment 3, fragment 4). It is due to our criterion that all fragments involved in multiple predictions have to span different regions of the MSA. d) Predictions with gapps. Let's assume that the full-length putative protein of the target gene model spans the whole alignment length. Having predicted (fragment 1, fragment 2) and (fragment 1, fragment 3) as splits, we can observe that when taken as a union, fragments 1, 2 and 3 do not span the whole MSA. However, we proceed with accepting the predictions and classify them as unambiguous.

3.2.6 Datasets

As a test case for evaluation and application of the methods, we used putative protein sequences of gene models constructed on the genome assembly of bread wheat, i.e. *Triticum aestivum* cv. Chinese Spring. A highly fragmented chromosome-by-chromosome survey sequence (IWGSP1 assembly, 2013-11-MIPS gene models) (International Wheat Genome Sequencing Consortium (IWGSC), 2014) and a high-quality reference sequence of chromosome 3B (Choulet *et al.*, 2014) provide a good basis to evaluate our methodology on a challenging dataset. Each putative gene was represented by one putative protein sequence in the proteome datasets. Putative wheat proteome contained 90,895 protein sequences of which 5,609 were assigned to chromosome 3B while the high-quality 3B dataset contained 6,033 putative protein sequences (please see section 1.3.2 for more information on the assemblies and annotation).

3.2.7 Recall on artificially fragmented datasets

To determine the recall of the methods, i.e. the proportion of fragmented gene models that the methods can identify (3.1) (Kent *et al.*, 1955), we simulated fragmentation on putative protein sequences of gene models assigned to a high-quality assembly of bread wheat chromosome 3B (3B reference sequence) (Choulet *et al.*, 2014).

$$recall = \frac{\#True\ Positive\ predictions\ (TP)}{\#True\ Positive\ predictions\ (TP) + \#False\ Negative\ predictions\ (FN)} \quad (3.1)$$

All putative protein sequences and their putative families were obtained from Ensembl Plants (Cunningham *et al.*, 2019; Howe *et al.*, 2020), release 31. We randomly chose one hundred sequences, each at least 100 amino acids long, and split them at a random position such that both fragments were at least 50 amino acids long (Fig. 3.6). All alignments were performed using Mafft v7.164b (Katoh and Standley, 2013) with default parameters. Protein

trees were built by FastTree v2.1.8 (Price, Dehal and Arkin, 2010), also with a default set of parameters. Obtained predictions correspond to the number of True Positive predictions (TP) while the ones we did not manage to recover represent the number of False Negative predictions (here: $\#FN = 100 - \#TP$) in formula (3.1).

In addition, we simulated fragmentation in a more challenging setting, i.e. on small putative protein families typically containing only sequences from evolutionarily very close paralogs (according to estimations). As a source of putative homologous groups, we used top-level hierarchical orthologous groups (HOGs). They were computed by the GETHOGs algorithm (Altenhoff *et al.*, 2013) with a default set of parameters on the input dataset comprised of putative proteomes of thirteen plants: bread wheat and twelve flowering plants exported from OMA Browser (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018) (Table B.1).

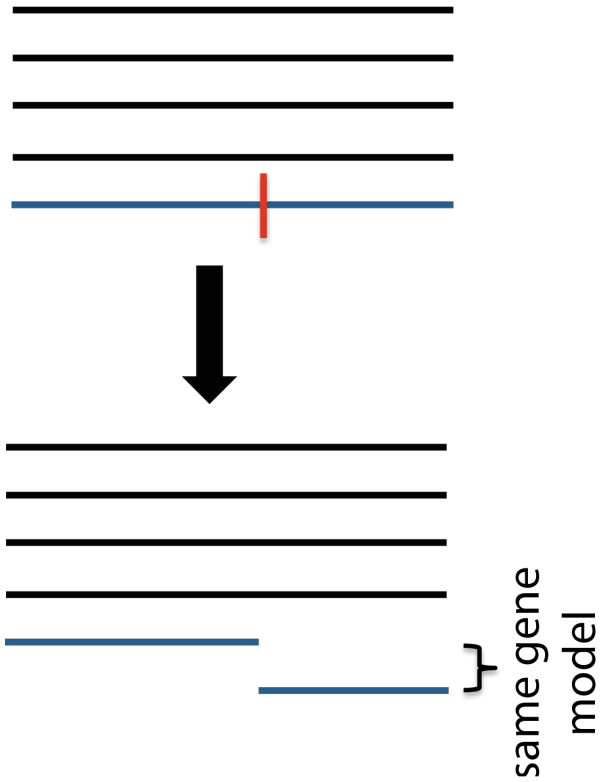


Figure 3.6: Simulating fragmentation—fragments coming from the same gene model.

First, we aligned corresponding putative protein families using Mafft v7.164b (Katoh and Standley, 2013) with default settings. Then we chose a random position *n* in the alignment such that the protein sequence of the target gene model, i.e. the one we want to fragment, contains at least 50 amino acids in the first *n* positions of the alignment and at least 50 amino acids right of the chosen position. First *n* positions of the aligned target protein sequence extended by *alignment_length-n* gaps form one fragment, while the second fragment is formed from *n* gaps extended by the rest of aligned target sequence. The original target sequence is then replaced with newly formed fragments while the rest of the putative family is kept the same.

3.2.8 Precision on artificially fragmented datasets

Precision is another common measure of the quality of methods. It penalises for erroneous predictions by measuring the proportion of correct predictions among all predictions that are made (3.2) (Kent *et al.*, 1955).

$$precision = \frac{\#True\ Positive\ predictions\ (TP)}{\#True\ Positive\ predictions\ (TP) + \#False\ Positive\ predictions\ (FP)} \quad (3.2)$$

Thus, here it measures the proportion of predictions that are indeed fragmented gene models.

To inspect cases where the methods incorrectly predict split gene models, we simulated fragments from pairs of putative paralogs assigned to the bread wheat 3B reference sequence (Choulet *et al.*, 2014) using the same datasets as above. We chose putative protein sequences of one hundred pairs of same-species gene models inferred as paralogous, cut them at a random position and took two complementary fragments (one from each initial sequence) each being at least 50 amino acids long (Fig. 3.7). Again, MSAs were obtained by Mafft v7.164b (default parameters) (Katoh and Standley, 2013) and protein trees by FastTree v2.1.8 (default parameters) (Price, Dehal and Arkin, 2010). Predictions obtained on these paralogous candidate fragments are False Positive predictions (FP) in formula (3.2). For the number of True Positive predictions (TP), we used the one obtained previously as explained in section 3.2.7.

Similarly as in section 3.2.7, we also simulated more challenging cases of fragmentation. We used the same set of HOGs as in the previous section.

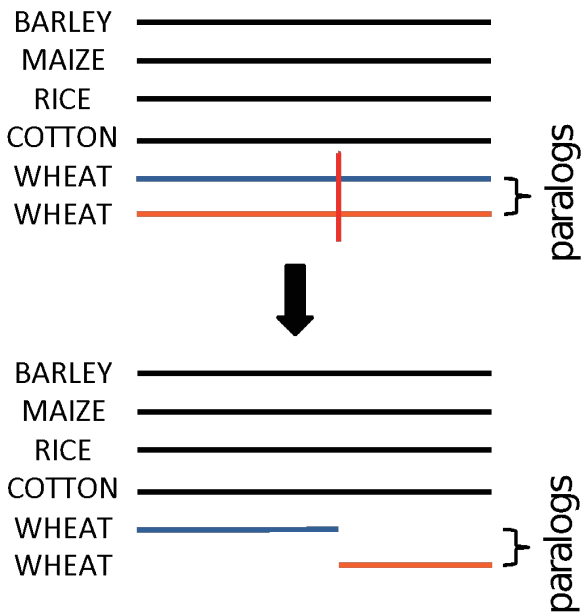


Figure 3.7: Simulating fragmentation—fragments coming from inferred paralogs.

We aligned putative protein families using Mafft v7.164b (Katoh and Standley, 2013) with default settings. Putative protein sequences of a randomly chosen pair of putative paralogs were assigned to *sequence 1* and *sequence 2* at random. Then we chose a random *n* such that the first *n* positions of *sequence 1* and last *alignment_length-n* positions of *sequence 2* each contain at least 50 amino acids. These two subsequences form the basis of simulated fragments, one extended by gaps on its right end and the other extended by gaps at its left end. If there was no such *n*, the pair was discarded.

3.2.9 Validation on low-quality assembly of bread wheat chromosome 3B

To assess predictions on the real data containing fragmented gene models, we applied our approaches to the putative protein sequences of 2013-11-MIPS gene models on the IWGSP1 low-quality assembly of bread wheat chromosome 3B—"3B survey sequence" (International Wheat Genome Sequencing Consortium (IWGSC), 2014), and compared the predictions with

the putative protein sequences of gene models on high-quality assembly of chromosome 3B ("3B reference sequence") (Choulet *et al.*, 2014) downloaded from URGI platform (URGI, 2009). As gold standard, we mapped sequences between the two assemblies using BLAST+ v2.2.30 (Camacho *et al.*, 2009).

For the predictions, we used the same reference species as in the simulations on HOGs (see sections 3.2.7-3.2.8) and the data was again exported from OMA Browser (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018) (Table B.2[38]). We computed protein families using the GETHOGs algorithm (Altenhoff *et al.*, 2013) with a default set of parameters. We generated 500 bootstrap samples for each top-level HOG and performed both tests on fragments overlapping less than 10% in the corresponding multiple sequence alignment. Sequences were aligned with Mafft v7.164b (default parameters) (Katoh and Standley, 2013) and trees built with FastTree v2.1.8 (default parameters) (Price, Dehal and Arkin, 2010) as above.

Since the GETHOGs algorithm (Altenhoff *et al.*, 2013) was not developed for the purpose of surveying genome assemblies and annotations, its default parameters might not be optimal for this purpose. In particular, a set of default parameters might be too conservative so we also computed HOGs with a different set of parameters (`MinScore := 150, LengthTol := 0.4, ReachabilityCutoff := 0.3`) which yielded bigger putative protein families and hence more candidates to test. We performed the tests with the same parameters as above on the deepest hierarchical orthologous groups.

For the assessment, the mapping of sequences between the survey and high-quality putative proteomes was not straightforward because the two differ not only in the degree of fragmentation, but also in some of the

[38] The OMA Browser (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018) release containing 3B survey sequence is older than the one containing 3B reference sequence. Hence, assemblies and annotations for some reference species differ between the releases as can be noticed in Tables B.1-B.2.

sequences themselves due to sequencing error, contamination, etc. To allow for a bit of tolerance while still maintaining unambiguous mapping between the two, we required hits to cover at least 95% of the corresponding query, the percentage identity in these matching regions to be at least 95%, and the hit to be unambiguous. As a stringent control, we performed a validation where, in addition to these two requirements, we only allowed mismatches to occur at the ends of a query sequence. The details for both assessments are provided in section B.3.

Since the set of fragmented gene models in the low-quality assembly is unknown and the assessment is further challenged by the outlined differences between the assemblies, it was not possible to calculate absolute recall rates for the tests. So we assessed the methods using the following values:

1) The number of predictions that could be mapped and verified as correct—as outlined above.
2) Precision of the tests based on the aforementioned mapping and verification.
3) Recall on the subset of gene models for which there is an indication that could be split. More details on the procedure are provided in Appendix B, section B.6.1.

3.2.10 Comparison to established methods

As a point of comparison, we employed the Ensembl Compara pipeline (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020) and ESPRIT (Dessimoz *et al.*, 2011) on the same 3B survey sequence as above (International Wheat Genome Sequencing Consortium (IWGSC), 2014). Again, the obtained predictions from each method were mapped to the 3B reference sequence (Choulet *et al.*, 2014) by BLAST+ v2.2.30 (Camacho *et al.*, 2009) to inspect if predicted pairs belong to the same gene model or not, requiring both coverage and percentage identity to be at least 95%. Validated predictions were compared to the results from validation

experiment on 3B survey sequence with the same BLAST+ criteria (described in section 3.2.9).

To obtain a comparable set of predictions on the 3B survey sequence (International Wheat Genome Sequencing Consortium (IWGSC), 2014) using public results available from the Ensembl Compara pipeline (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020), we filtered "gene_split" pairs from their homologies file (release 21). We took only pairs where both putative protein sequences of gene models were at least 50 amino acids long and such that, when corresponding putative protein family was aligned with Mafft v7.164b (Katoh and Standley, 2013), candidates overlapped for less than 10%. We also included cases where more than two gene models were inferred as a part of the same gene model given that no two putative protein sequences involved overlapped for 10% or more. Since some of the putative sequences could not be found in the OMA Browser (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018) dataset used for validating the collapsing and LRH approach, we classified Ensembl predictions into two groups: those that could be found in the OMA Browser dataset, and hence, included in the comparison, and those that could not.

Another set of predictions was obtained by running ESPRIT (Dessimoz *et al.*, 2011) on the same 3B survey sequence data (International Wheat Genome Sequencing Consortium (IWGSC), 2014) using twelve reference putative plants (the same dataset as in the section 3.2.9, Table B.2) keeping all default parameters but increasing the required length of the putative protein sequences of candidate gene models to be at least 50 amino acids (`MinSeqLenContig := 50`). We only considered a confident unambiguous set of predictions (`hits.txt` file).

3.2.11 ECOMB

We also considered a meta-approach ECOMB—encompassing ESPRIT (Dessimoz *et al.*, 2011) and the new combined approach. It takes the union of predictions made by ESPRIT and our joint method (collapsing branches with support lower than 0.95 and LRH with significance of 0.01).

$$ECOMB = ESPRIT \cup (collapsing\ @\ 0.95 \cap LRH\ @\ 0.01)$$

3.2.12 Application to putative bread wheat genome

Finally, we employed the tests to infer fragmented gene models in the first draft release of the whole putative bread wheat genome *Triticum aestivum* cv. Chinese Spring (IWGSP1, 2013-11-MIPS) (International Wheat Genome Sequencing Consortium (IWGSC), 2014). We considered only candidate fragments assigned to the same chromosome and the same chromosome arm. We used the same reference putative proteomes as in the previous analyses with HOGs in section 3.2.9. Based on simulations and validation on 3B survey sequence, we determined a set of parameters used for predictions. In particular, we ran GETHOGs (Altenhoff *et al.*, 2013) with default parameters and allow candidate fragments to mutually overlap less than 10% in the corresponding MSA of the top-level HOG. We used Mafft v7.164b (Katoh and Standley, 2013) to get alignments and FastTree v2.1.8 (Price, Dehal and Arkin, 2010) to construct trees, both with their default set of parameters. Finally, we chose 0.95 as a threshold for collapsing and set the significance of the LRH to 0.01.

3.2.13 Beyond the FastTree default settings

To explore the effect of tree reconstruction tools and their parameters to the outcomes of phylogenetic heuristics, we employed FastTree v2.1.7 default installation (Price, Dehal and Arkin, 2010), FastTree v2.1.10 double-precision installation and RAxML v8.2.12 (Alexandros Stamatakis, 2014). FastTree double-precision installation aims to resolve short branch lengths more accurately which is beneficial for families containing nearly-identical putative sequences. RAxML is widely used in the community[39] and performs more exhaustive tree search than FastTree. It could provide more reliable trees and it could be a preferable tree building tool of the potential users. Thus, it is important to quantify its effect on split gene model predictions made by our heuristics. All parameters of the tools used in the analysis are specified below.

FastTree default installation (Price, Dehal and Arkin, 2010):

- `-pseudo`: recommended for datasets with many fragmented sequences
- `-mlacc 2 -slownni`: more rounds of maximum-likelihood nearest-neighbor interchanges (NNIs) in a tree search
- `-spr 4`: more rounds of minimum-evolution subtree-prune-regraft (SPR) moves in a tree search
- `-mlacc 2 -slownni -spr 4`
- `-wag`: WAG (Whelan and Goldman, 2001) instead of default JTT model of amino acid evolution (Jones, Taylor and Thornton, 1992)
- `-gamma`: when a tree is reconstructed, rescale it to optimise the likelihood under gamma model with 20 rate categories (Yang, 1994)

FastTree double-precision installation (Price, Dehal and Arkin, 2010):

[39] As of 7 December 2019, RAxML-VI-HPC (Alexandros Stamatakis, 2006) was cited 13,449 times and RAxML v8 (A. Stamatakis, 2014) 11,520 times according to Google Scholar (Google, 2004). In comparison, FastTree (Price, Dehal and Arkin, 2009, 2010) was cited 6,630 times in total.

- default: CAT model of rate heterogeneity (A. Stamatakis, 2006; Stamatakis, 2016)[40] with 20 rate categories, JTT substitution model (Jones, Taylor and Thornton, 1992)
- `-lg`[41]: LG (Le and Gascuel, 2008) instead of default JTT model of amino acid evolution
- `-pseudo`
- `-mlacc 2 -slownni`
- `-spr 4`
- `-mlacc 2 -slownni -spr 4`
- `-wag`
- `-gamma`
- `-pseudo -mlacc 2 -slownni -spr 4`

Last 7 settings have the same parameter interpretation as in the default installation version of the tool (see above).

RAxML (Alexandros Stamatakis, 2014):

- `-m PROTCATJTT -c 20`: CAT model of rate heterogeneity (A. Stamatakis, 2006; Stamatakis, 2016)[42] with 20 rate categories, JTT substitution model—as default FastTree parameters
- `-m PROTCATLG -c 20`: CAT model of rate heterogeneity with 20 rate categories, LG substitution model (Le and Gascuel, 2008)
- `-m PROTGAMMAAUTO`: gamma model of rate heterogeneity (Yang, 1994, 1996), RAxML determines the best substitution model for the data among twenty of them
- `-m PROTGAMMAGTR`: gamma model of rate heterogeneity, GTR substitution model (Rodríguez *et al.*, 1990)

Other steps in the pipeline remained the same as described in sections 3.2.7-3.2.8. Furthermore, we used the same input dataset with 200 putative wheat sequences assigned to chromosome 3B, artificially fragmented and

[40] Note that this is not a CAT model of Lartillot and Philippe (2004).
[41] The option was not available in the FastTree default installation used in this study.
[42] Note that this is not a CAT model of Lartillot and Philippe (2004).

placed in putative homologous protein families with twelve other putative plant proteomes (Table B.1) using GETHOGs algorithm (Altenhoff *et al.*, 2013).

Note that due to the change of computational resources, here we employed different releases of FastTree (Price, Dehal and Arkin, 2010) than in the rest of the chapter where FastTree v2.1.8 was used. For computations with default installation, here we used FastTree v2.1.7 which does not differ from v2.1.8 in terms of tree reconstruction algorithms (Arkin Lab, 2008; Dehal *et al.*, 2010). FastTree v2.1.10 double-precision release provides an option `-lg` not available in v2.1.7 and v2.1.8 but no methodological changes affecting the computations of our interest were introduced in the meantime.

3.3 Results

Recall that we aim to identify fragments derived from the same gene but wrongly annotated as separate gene models in a putative genome of interest, leveraging putative genomes of related species (or more precisely, corresponding putative proteomes). In the previous section (3.2), we introduced two heuristic phylogenetic methods: a heuristic relying on collapsing branches with low SH-like support and likelihood ratio heuristic (LRH). To evaluate the heuristics and determine parameters for predictions on the putative bread wheat genome, we took three approaches. First, we simulated fragmentation on the real data to calculate recall and precision. Then we applied both methods to the bread wheat chromosome 3B survey sequence (IWGSP1, 2013-11-MIPS) (International Wheat Genome Sequencing Consortium (IWGSC), 2014) and validated predictions with respect to the 3B reference sequence (Choulet *et al.*, 2014). We used the same wheat data and validation approach to compare the predictions to the ones obtained by established methods, namely Ensembl Compara pipeline (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020) and ESPRIT (Dessimoz *et al.*, 2011). Finally, based on the best parameters

obtained from these analyses, we applied the method to infer split gene models in other 20 putative chromosomes of the survey wheat IWGSP1 assembly and its 2013-11-MIPS annotation.

3.3.1 Simulated fragmentation: moderate recall of the collapsing heuristic, LRH more successful

To assess the recall of the methods, we retrieved putative protein families inferred by Ensembl Compara (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020), simulated fragmentation in 100 putative protein sequences identified on the high-quality wheat 3B reference assembly (Choulet *et al.*, 2014) and tried to recover these pairs. On these challenging simulations, the collapsing approach yielded moderate recall ranging from 0.20 to 0.58, while the LRH demonstrated ability to recover split gene models with recall between 0.81 and 0.99 (Fig. 3.8a, Table B.3). We also evaluated an approach that combines the two methods. A split gene model was inferred if both methods were in agreement. As expected, this approach resembled the recall of the collapsing approach with the same threshold.

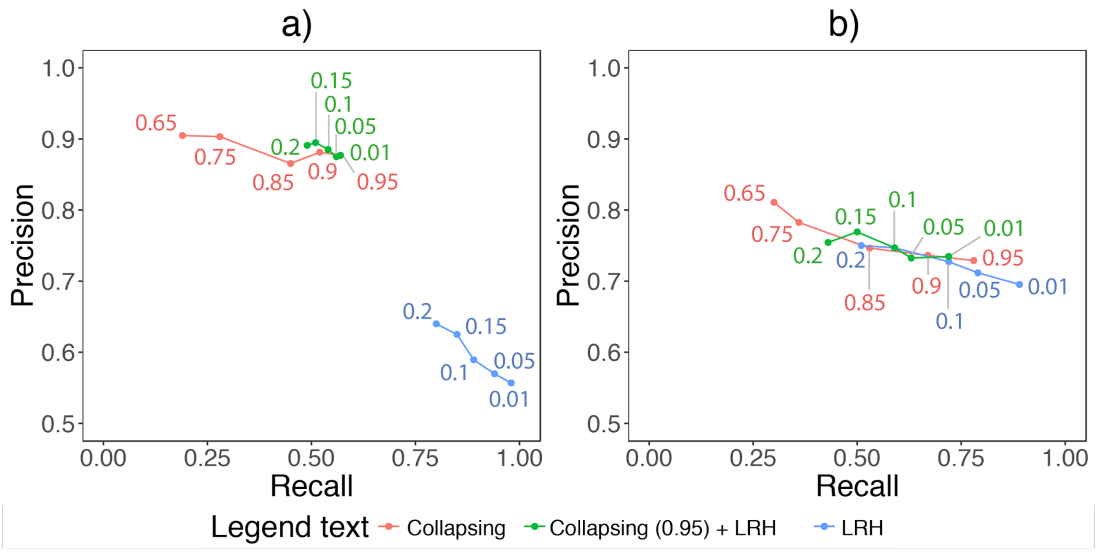


Figure 3.8: Precision and recall of the methods on artificially fragmented putative protein sequences of gene models constructed on high-quality putative wheat chromosome 3B (Choulet *et al.*, 2014).

a) Simulated fragmentation on Ensembl putative protein families (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020), b) Simulated fragmentation on HOGs with default settings in GETHOGs algorithm (Altenhoff *et al.*, 2013). We performed the collapsing approach with a set of thresholds {0.65, 0.75, 0.85, 0.9, 0.95}, likelihood ratio heuristics (LRH) with a set of significance levels {0.2, 0.15, 0.1, 0.5, 0.01}, and the combination of the collapsing approach (threshold 0.95) and LRH (thresholds {0.2, 0.15, 0.1, 0.5, 0.01}). The collapsing approach yielded moderate recall, lower than the LRH, while its precision was higher than that of the LRH.

As a control, we performed another set of simulations using a different set of input putative homologs—deepest OMA hierarchical orthologous groups (HOGs) (Altenhoff *et al.*, 2013) containing putative protein sequences from thirteen plants including wheat. Recall of the collapsing approach varied between 0.30 and 0.78 and the recall of the LRH was between 0.51 and 0.89 (Fig. 3.8b, Table B.4). The combined approach, unsurprisingly, resembles the collapsing approach with the same threshold.

Supplementary files with gene model IDs, cut positions and outcomes of the heuristic inference for all cases can be downloaded from https://doi.org/10.6084/m9.figshare.11734467.v1.

3.3.2 Simulated fragmentation: moderate precision of the LRH, the collapsing approach attains higher

To compute precision of the heuristics, in addition to one hundred fragmented gene models, we also included one hundred pairs of non-overlapping fragments generated from putative paralogs—which can be very difficult negative cases if the paralogs are near-identical at the protein sequence level. On the data coming from Ensembl (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020), collapsing yielded precision ranging from 0.85 to 0.88, while the LRH yielded precision in the interval between 0.56 and 0.64 (Fig. 3.8a, Table B.3). An approach combining two methods demonstrated slightly higher precision than the collapsing approach.

In the control experiments on the OMA HOGs (Altenhoff *et al.*, 2013), precision with the collapsing method was between 0.73 and 0.81, while precision with the LRH scored between 0.70 and 0.75 (Fig. 3.8b, Table B.4). The combined approach was equally or slightly more precise than the collapsing approach with the same threshold for collapsing branches.

Supplementary files with gene model IDs, cut positions and outcomes of the heuristic inference for all cases are available at https://doi.org/10.6084/m9.figshare.11734467.v1.

3.3.3 Low-quality fragmented data: methods perform with higher precision, ability to identify fragmentation remains consistent

To further assess the approaches and identify suitable parameters, we applied our methods to the chromosome 3B of the bread wheat survey genome (IWGSP1, 2013-11-MIPS) (International Wheat Genome Sequencing Consortium (IWGSC), 2014). This is the one chromosome for which a high-quality reference was available (Choulet *et al.*, 2014) but which was not used for creating the draft whole-genome assembly.

Overall, the methods achieved higher precision than when applied to simulated fragmentation (Fig. 3.9a, Table B.5). The analysis showed particularly high precision of the collapsing approach. The absolute recall rate could not be easily assessed on these real data as not every gene model from the putative survey genome could be uniquely mapped (or mapped at all) to a gene model in the putative reference genome. So we considered 1) the number of predictions that could be mapped and verified as correct (Fig. 3.9), and 2) recall on the subset of gene models for which we found indications that could be split (Fig. 3.10). Obtained results were consistent with the simulations (Fig. 3.9, Fig. 3.10a-c).

One challenge with this setup was the fact that the draft survey sequence assembly contains other types of problems, such as sequencing errors or ~10% contamination from other chromosomes. If we only consider fragments that can be perfectly mapped between gene models on the draft whole-genome assembly and the reference assembly (no mismatch in their central part, see section B.3.2), the number of predictions that could be validated diminishes, but on the remaining set, our approaches showed even higher precision (Fig. 3.9c, Table B.6), indicating that the performance reported in Figure 3.9a is conservative.

The control experiments also gave consistent results (Fig. 3.9b, Fig. 3.9.d, Tables B.7-B.8). Due to relaxed parameters in HOGs inference (Altenhoff *et al.*, 2013), the putative protein families were larger and provided more

candidates for heuristic inference. However they contained putative sequences coming from more distant genes[43], some of which were wrongly assigned to families, making it a more challenging setting for phylogenetic inference. As expected, the number of predicted split gene models increased, but at a cost of lower precision.

The approximations to recall values obtained on the cases subjected to heuristic inference (Fig. 3.10a-c, Tables B.10-B.13) were consistent with recall on simulated fragmentation experiments (Fig. 3.8). Generally speaking, we obtained higher estimated recall values with relaxed parameters in the GETHOGs algorithm (Altenhoff *et al.*, 2013) combined with stringent BLAST+ (Camacho *et al.*, 2009) pair mapping. This is because, as explained above, these starting families contained more candidate pairs. The downsides of working with larger and more challenging putative families were at least partially balanced out by considering only stringent mappings between putative sequences. A similar trend is observable when starting from the smaller families obtained with default GETHOGs settings: there too, more stringent mapping led to higher recall estimates.

Next, we sought to better understand the source of false negative predictions. Surprisingly, we found that the main issue was *upstream* of our tests: only ~11.5-28.3% of the pairs with common best mapping were subjected to our heuristics (Table B.9). Some pairs did not meet our criteria to qualify as candidate pairs because fragments were too short (< 50 AA) or had long mutual overlap (≥ 10% of their lengths) in the multiple sequence alignment. But the main reason, strikingly, was that most pairs were not found in the same input family (~67-88.3% of cases not subjected to the heuristics). This happened because the GETHOGs method (Altenhoff *et al.*, 2013) which we used to compute the input putative homologous groups was designed to cluster putative sequences for which indications of homology can be found over most of their length—and that was not the case here

[43] According to estimation in HOGs pipeline (Altenhoff *et al.*, 2013)

(section B.6.3). Consequently, when the number of predicted splits was divided by the number of all pairs where both putative sequences had common mapping, the result was quite low (Fig. 3.10d-f). This motivated us to merge HOGs containing potential fragments of the same gene model and proceed with testing (section B.6.4). The heuristics found indications for more fragmented gene models (Tables B.15-B.16) yet many pairs remained unexamined because the putative sequences could not be found in any HOG as they were discarded by GETHOGs algorithm.

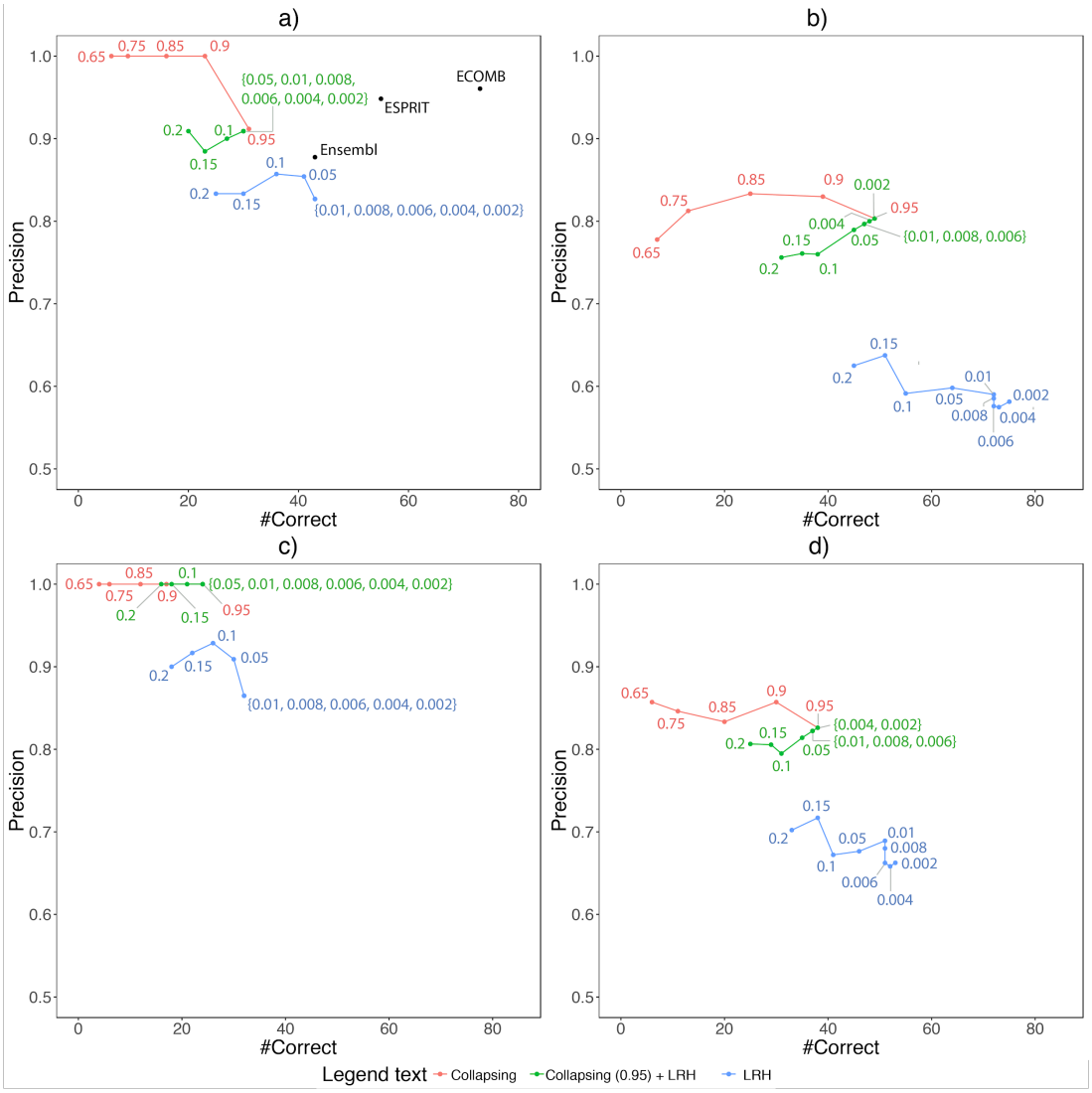


Figure 3.9: Validation on gene models of low-quality bread wheat chromosome 3B assembly (International Wheat Genome Sequencing Consortium (IWGSC), 2014).

a) Split gene models inferred on the low-quality ("survey") putative wheat chromosome 3B using HOGs with default parameters in GETHOGs algorithm (Altenhoff *et al.*, 2013), and validated against putative high-quality wheat 3B (Choulet *et al.*, 2014) using BLAST+ (Camacho *et al.*, 2009) with less stringent criteria. The figure also includes comparison with three other approaches (Ensembl Compara (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020), ESPRIT (Dessimoz *et al.*, 2011) and ECOMB). ECOMB combines ESPRIT's and the predictions inferred when combining the collapsing approach (threshold 0.95) and LRH (significance 0.01). b) Validation on 3B survey sequence using HOGs with relaxed parameters in

GETHOGs algorithm, less stringent BLAST+ validation, c) Validation on 3B survey sequence using HOGs with default settings in GETHOGs algorithm, more stringent BLAST+ validation, d) Validation on 3B survey sequence using HOGs with relaxed parameters in GETHOGs algorithm, more stringent BLAST+ validation. We performed the collapsing approach with a set of thresholds {0.65, 0.75, 0.85, 0.9, 0.95}, likelihood ratio heuristics (LRH) with a set of significance levels {0.2, 0.15, 0.1, 0.5, 0.01, 0.008, 0.006, 0.004, 0.002}, and the combination of the collapsing approach (threshold 0.95) and LRH (thresholds {0.2, 0.15, 0.1, 0.5, 0.01, 0.008, 0.006, 0.004, 0.002}). Overall, the approaches showed higher precision than on the simulated fragmentation (Fig. 3.8). The recall could not be established so we used the number of correct predictions as a surrogate.

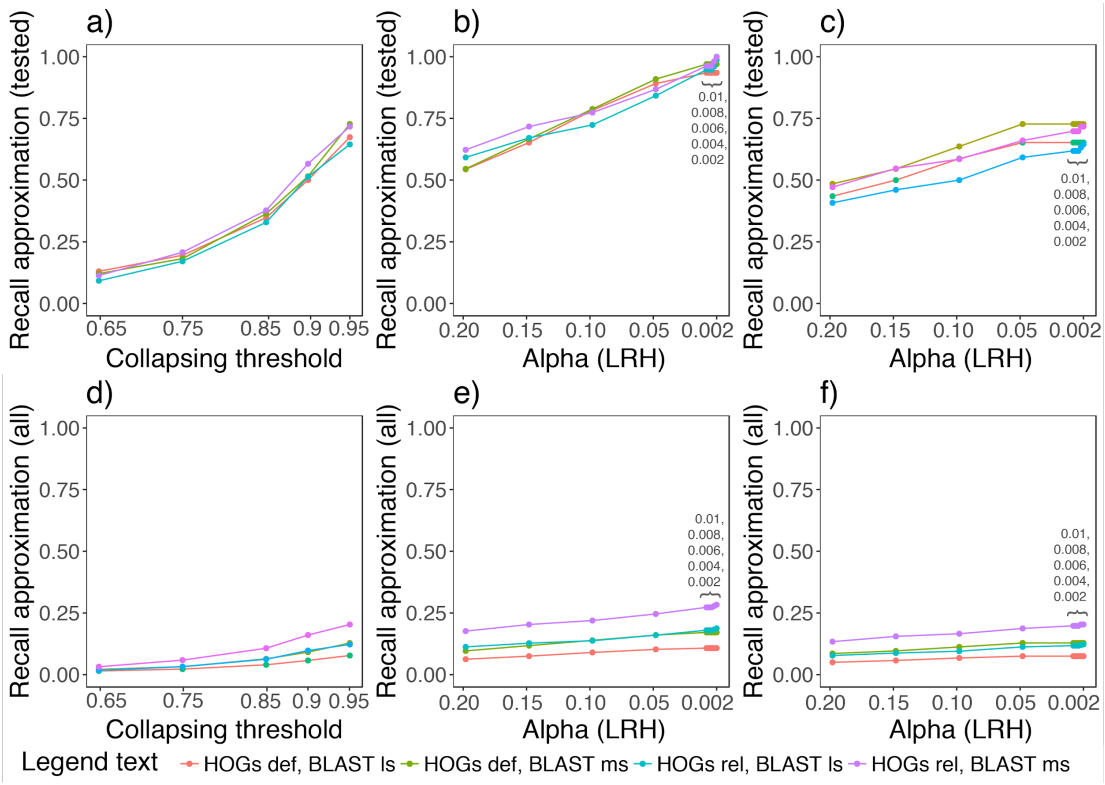


Figure 3.10: Validation on gene models of low-quality bread wheat chromosome 3B assembly (International Wheat Genome Sequencing Consortium (IWGSC), 2014) for which there is an indication that could be fragmented—approximation to recall values.

a) Recall of the collapsing approach on the cases subjected to the heuristic inference[44]. b) Recall of the likelihood ratio heuristic (LRH) on the cases subjected to the inference. c) Recall of the combined approach (collapsing with threshold 0.95 + LRH) on the cases subjected to the heuristic inference. d) Recall of the collapsing approach on all cases for which we found indications that could be fragments. e) Recall of the LRH on all cases for which we found indications that could be fragments. f) Recall of the combined approach (collapsing with threshold 0.95 + LRH) on all cases for which we found indications that could be fragments. Input protein families were obtained using HOGs with default (HOGs def) and relaxed parameters (HOGs rel) in GETHOGs algorithm (Altenhoff *et al.*, 2013), and validated against high-quality wheat 3B data (Choulet *et al.*, 2014) using BLAST+

[44] Some cases were not scrutinised by the heuristics as candidates were not found in the same HOG, were too short or overlapped too much in the corresponding multiple sequence alignment.

(Camacho *et al.*, 2009) with less (BLAST ls) and more stringent (BLAST ms) criteria.

Gene model IDs for predictions summarised in Figure 3.9 can be downloaded from https://doi.org/10.6084/m9.figshare.11733597.v1, while those depicted in Figure 3.10 are available at https://doi.org/10.6084/m9.figshare.11733609.v1.

3.3.4 Established methods show high precision and recall

To gain further insights into the performance of the proposed approaches, we compared them to two existing methods, namely Ensembl Compara pipeline (which however cannot easily run on custom genome data) (Vilella *et al.*, 2009;Cunningham *et al.*, 2019; Howe *et al.*, 2020) and ESPRIT (Dessimoz *et al.*, 2011). Both methods were applied to the 3B survey sequence (IWGSP1 assembly, 2013-11-MIPS gene models) (International Wheat Genome Sequencing Consortium (IWGSC), 2014) and then validated against the 3B reference sequence (Choulet *et al.*, 2014) using BLAST+ (Camacho *et al.*, 2009). In terms of the number of correct predictions, Ensembl Compara and ESPRIT performed equally well or better than our approaches displaying high precision (Fig. 3.9a and Table 3.1). Further analysis showed that predictions from different methods are rather complementary and worthwhile taking into account (Fig. 3.11).

Summary of predictions on the data that can be found in OMA					
	#total	#could not validate	#correct	#wrong	Precision
Ensembl Compara	86	47	33	6	0.85
ESPRIT	204	146	55	3	0.95
Summary of all Ensembl's unambiguous predictions					
Ensembl Compara	#total	#could not validate	#correct	#wrong	Precision
	106	57	43	6	0.88

Table 3.1: Performance of Ensembl Compara (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020) and ESPRIT (Dessimoz *et al.*, 2011).

Summary.

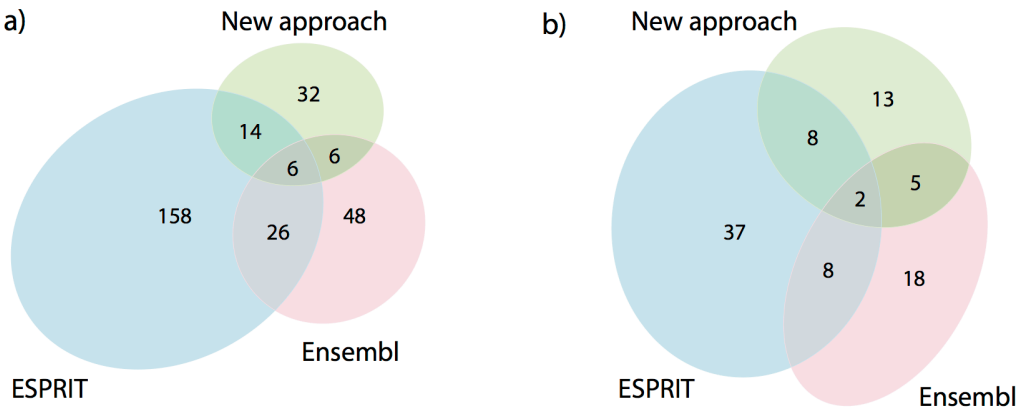


Figure 3.11: Comparison to Ensembl Compara (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020) and ESPRIT (Dessimoz *et al.*, 2011).

"New approach" denotes a combination of the collapsing approach (threshold 0.95) and LRH (significance 0.01). a) The number of predictions inferred by each method on 3B survey sequence (IWGSP1 assembly, 2013-11-MIPS gene models) (International Wheat Genome Sequencing Consortium (IWGSC), 2014), b) The number of predictions on 3B survey sequence

classified as correct in the less stringent BLAST+ validation (Camacho *et al.*, 2009). The three methods yielded mainly complementary predictions.

Gene model IDs involved in predictions alongside results of the validation can be retrieved from https://doi.org/10.6084/m9.figshare.11704266.v1.

3.3.5 Meta-approach: obtaining more predictions and with higher confidence

Given the complementarity of the predictions made by different methods (Fig. 3.11), we also considered a meta-approach, which we call ECOMB, comprising a union of predictions made by ESPRIT (Dessimoz *et al.*, 2011) and predictions resulting from a combination (intersection) of the collapsing approach (threshold 0.95) and LRH (significance 0.01). ECOMB inferred the biggest number of correct predictions with high precision (Fig. 3.9a and Table 3.2).

Summary of predictions on the data that can be found in OMA					
	#total	#could not validate	#correct	#wrong	Precision
ECOMB	242	166	73	3	0.96

Table 3.2: Performance of ECOMB = ESPRIT ∪ (collapsing @ 0.95 ∩ LRH @ 0.01).

Summary.

3.3.6 1,221 unambiguous predictions on the putative wheat genome

Finally, we applied our heuristics to infer split gene models on the rest of the putative bread wheat genome (IWGSP1 assembly, 2013-11-MIPS gene models) (International Wheat Genome Sequencing Consortium (IWGSC), 2014), i.e. all chromosomes other than 3B. Based on the analyses on

simulated fragmentation and between two assemblies (see sections 3.3.1-3.3.3), we determined parameters for the heuristics. For each putative chromosome arm, we obtained putative protein families by running OMA GETHOGs (Altenhoff *et al.*, 2013) with default parameters. In the collapsing approach, we collapsed all branches with SH-like support less than 0.95, and we applied the likelihood ratio heuristic with a significance level of 0.01. The intersection of predictions identified 1,442 pairs in total: 1,221 unambiguous and 221 ambiguous cases. The distribution of the number of predictions per chromosome is shown in Figure 3.12 (see also Table B.17) while fragment IDs can be downloaded from https://doi.org/10.6084/m9.figshare.11704257.

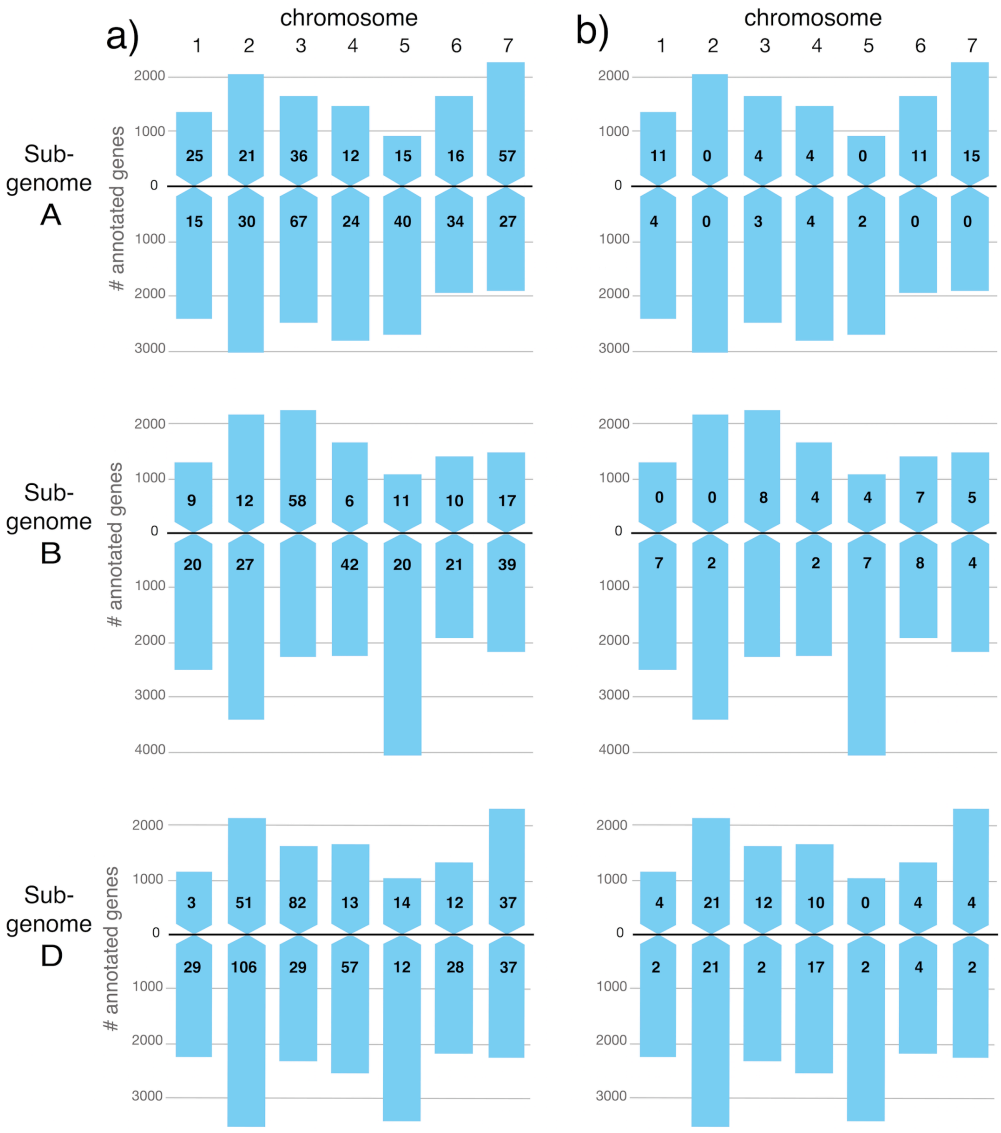


Figure 3.12: Inferred gene model splits on the putative bread wheat genome (IWGSP1 assembly, 2013-11-MIPS gene models) (International Wheat Genome Sequencing Consortium (IWGSC), 2014).

Allohexaploid genome (AABBDD; 6x=2n=42) is depicted in 3 parts, each representing one homeologous subgenome (see section 1.3.2 for more details). Each putative chromosome arm is plotted in size proportional to the number of (previously) annotated gene models in the input dataset (90,895 gene models in total). Numbers on chromosomes represent the number of our predictions. a) Number of unambiguous predictions for each putative chromosome arm. b) Number of ambiguous predictions (i.e. for which there is more than two candidate fragments for a single juncture). Pairs of

fragments were inferred separately for each putative chromosome arm of flow-sorted *Triticum aestivum* cv. Chinese Spring, except chromosome 3B, for which the analysis was performed on the entire putative chromosome. We identified a total of 1,221 unambiguous and 221 ambiguous cases.

3.3.7 On different tree reconstruction methods

Tree reconstructions with FastTree v2.1.7 default installation (Price, Dehal and Arkin, 2010), FastTree v2.1.10 double-precision installation and RAxML v8.2.12 (Alexandros Stamatakis, 2014) on 200 artificially fragmented putative protein sequences of wheat gene models annotated on chromosome 3B (Choulet *et al.*, 2014) and assigned to putative protein families with putative protein sequences from twelve other plants (Table B.1) using GETHOGs algorithm (Altenhoff *et al.*, 2013) with default settings—the same data as in sections 3.2.7 and 3.2.8—provided consistent results across different runs (Fig. 3.8b, Fig. 3.13-3.15, Table 3.3, Tables B.18-B.20).

Heuristics performed on trees computed with FastTree double-precision installation (Price, Dehal and Arkin, 2010) slightly outperformed heuristics on trees computed with FastTree default installation (Fig. 3.13-3.14, Table 3.3, Tables B.18-B.19) which was expected given the input putative protein families and the fact that double-precision version aims to calculate short branch lengths more accurately. While recall of the heuristics on FastTree default installation trees ranged from 0.3 to 0.92, it was between 0.35 and 0.92 when double-precision output trees were used. Similarly, precision of the heuristics on the FastTree default installation outputs was between 0.67 and 0.83 while it was between 0.68 and 0.86 using double-precision output trees. Based on the results of the combined approach where a threshold for the collapsing approach was set to 0.95 and likelihood ratio heuristic to 0.01, we would like to highlight the following FastTree options and corresponding results:

- `-pseudo`

Computations with this option provided trees which yielded the highest precision—0.747 among runs using FastTree default installation (corresponding recall 0.74) and 0.755 among runs using FastTree double-precision installation (corresponding recall 0.8).

- `-mlacc 2 -slownni`

  Runs with this option provided trees which led to the highest recall of the tests in the pipeline relying on the FastTree default installation—0.76 (corresponding precision 0.738). It also scored the second best recall in the setting where double-precision installation was employed—of 0.81 (and precision 0.75). Interestingly, combining this option with `-spr 4` or `-pseudo -spr 4` did not further improve neither recall nor precision on this particular dataset.

- `-lg`

  Reconstructed trees enabled the highest recall of the tests in the setting with double-precision FastTree installation—0.84 (precision 0.74). Unfortunately, the option was not available in the FastTree default installation used in this study.

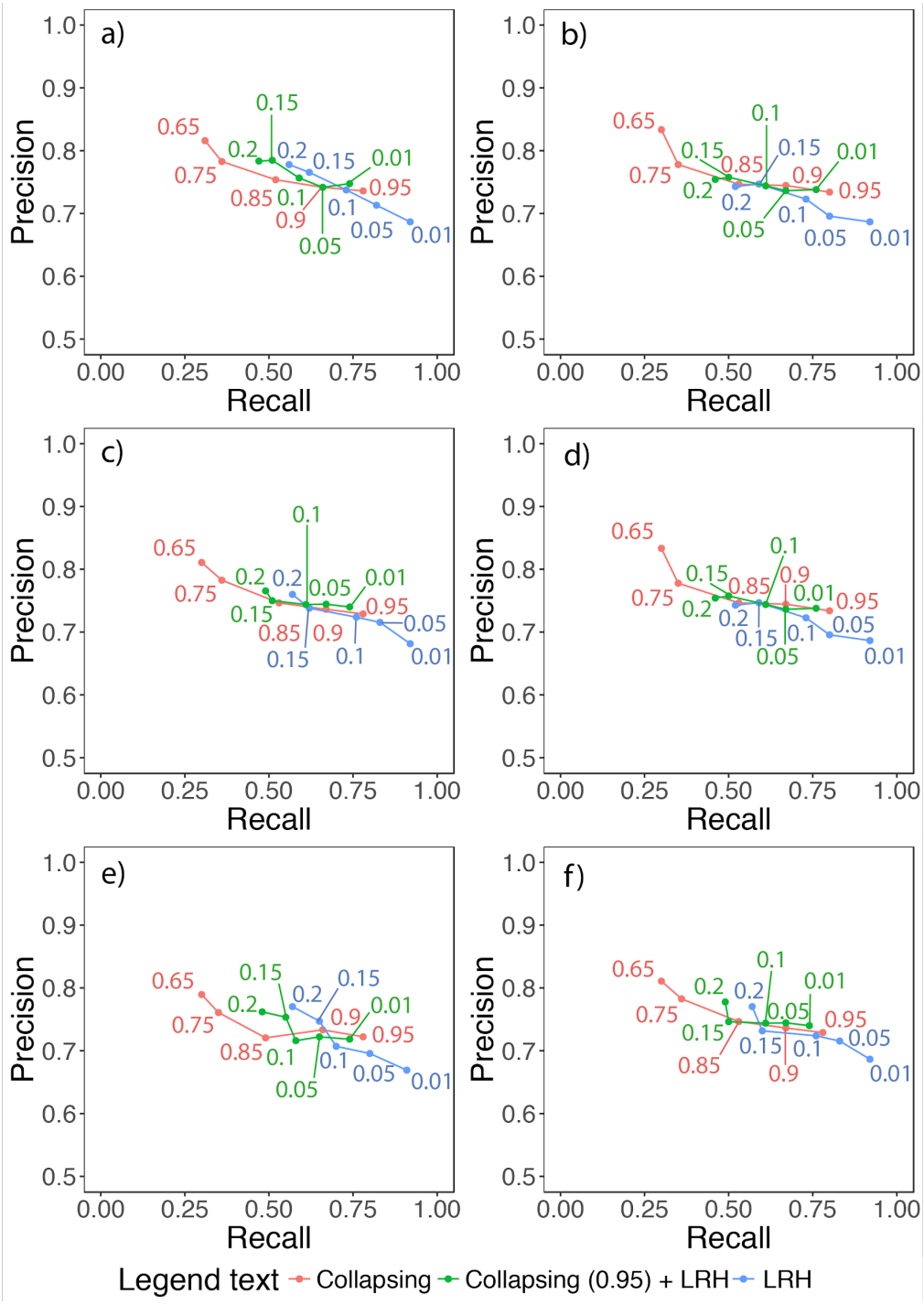


Figure 3.13: Exploring the effect of different FastTree parameters (default installation, v2.1.7) (Price, Dehal and Arkin, 2010): Precision and recall of the heuristics.

Artificially fragmented putative protein sequences constructed on putative high-quality assembly of wheat chromosome 3B (Choulet *et al.*, 2014), HOGs with default settings in GETHOGs algorithm (Altenhoff *et al.*, 2013) on the data listed in Table B.1. FastTree parameters: a) `-pseudo`, b) `-mlacc`

`2 -slownni`, c) `-spr 4`, d) `-mlacc 2 -slownni -spr 4`, e) `-wag`, f) `-gamma`. Just as before, we performed the collapsing approach with a set of thresholds {0.65, 0.75, 0.85, 0.9, 0.95}, likelihood ratio heuristics (LRH) with a set of significance levels {0.2, 0.15, 0.1, 0.5, 0.01}, and the combination of the collapsing approach (threshold 0.95) and LRH (thresholds {0.2, 0.15, 0.1, 0.5, 0.01}). We observed consistent outcomes of the heuristics across chosen parameters. The outcomes were also consistent with the performance of the heuristics on trees reconstructed using default parameters shown in Fig. 3.8b, and consistent with the performance of the heuristics when trees were reconstructed with FastTree double-precision (Fig. 3.14) or RAxML (Alexandros Stamatakis, 2014) (Fig. 3.15).

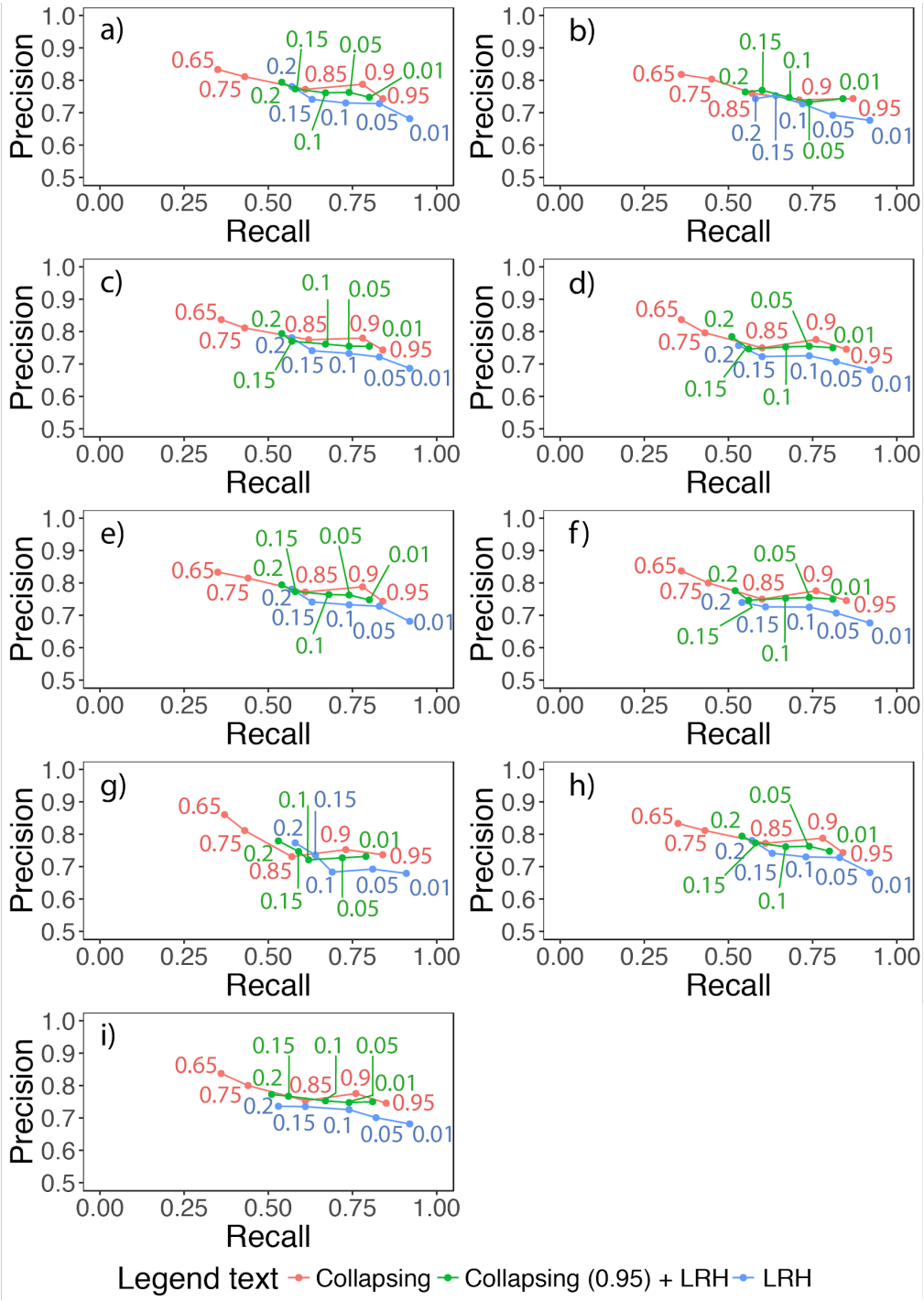


Figure 3.14: Exploring the effect of different FastTree parameters (double-precision installation, v2.1.10) (Price, Dehal and Arkin, 2010): Precision and recall of the tests.

Artificially fragmented putative protein sequences constructed on putative high-quality wheat chromosome 3B (Choulet *et al.*, 2014), HOGs with default settings in GETHOGs algorithm (Altenhoff *et al.*, 2013) on the data listed in Table B.1. FastTree parameters: a) default, b) `-lg`, c) `-pseudo`, d) `-mlacc`

`2 -slownni`, e) `-spr 4`, f) `-mlacc 2 -slownni -spr 4`, g) `-wag`, h) `-gamma`, i) `-pseudo -mlacc 2 -slownni -spr 4`. Again, we performed the collapsing approach with a set of thresholds {0.65, 0.75, 0.85, 0.9, 0.95}, the likelihood ratio heuristics (LRH) with a set of significance levels {0.2, 0.15, 0.1, 0.5, 0.01}, and the combination of the collapsing approach (threshold 0.95) and LRH (thresholds {0.2, 0.15, 0.1, 0.5, 0.01}). We observed consistent outcomes of the heuristics regardless of the choice of parameters, FastTree installation (default installation led to performance depicted in Fig. 3.8b and Fig. 3.13) and a choice of tree-reconstruction tool (Fig. 3.15 for heuristics on RAxML (Alexandros Stamatakis, 2014) trees).

On this particular dataset with a given set of tested installations and parameters for FastTree (Price, Dehal and Arkin, 2010) and RAxML (Alexandros Stamatakis, 2014), trees reconstructed by RAxML led to the heuristics performance comparable with the performance when the trees were reconstructed by FastTree default installation, and was slightly outperformed by the outcomes of the pipeline employing FastTree double-precision installation (Fig. 3.13-3.15, Table 3.3, Tables B.18-B.20). Recall of the heuristics performed on RAxML output trees was between 0.23 and 0.9 while precision scored between 0.7 and 0.8. Among tested options, `-m PROTGAMMAGTR` and `-m PROTCATJTT -c 20` (as default FastTree parameters[45]) generally led to the highest precision and recall of the heuristics. The combined approach with collapsing threshold of 0.95 and likelihood ratio heuristic at 0.01 achieved the highest recall and precision on the trees from `-m PROTGAMMAGTR` run—0.81 and 0.77 respectively.

[45] FastTree uses the same “CAT” model (A. Stamatakis, 2006).

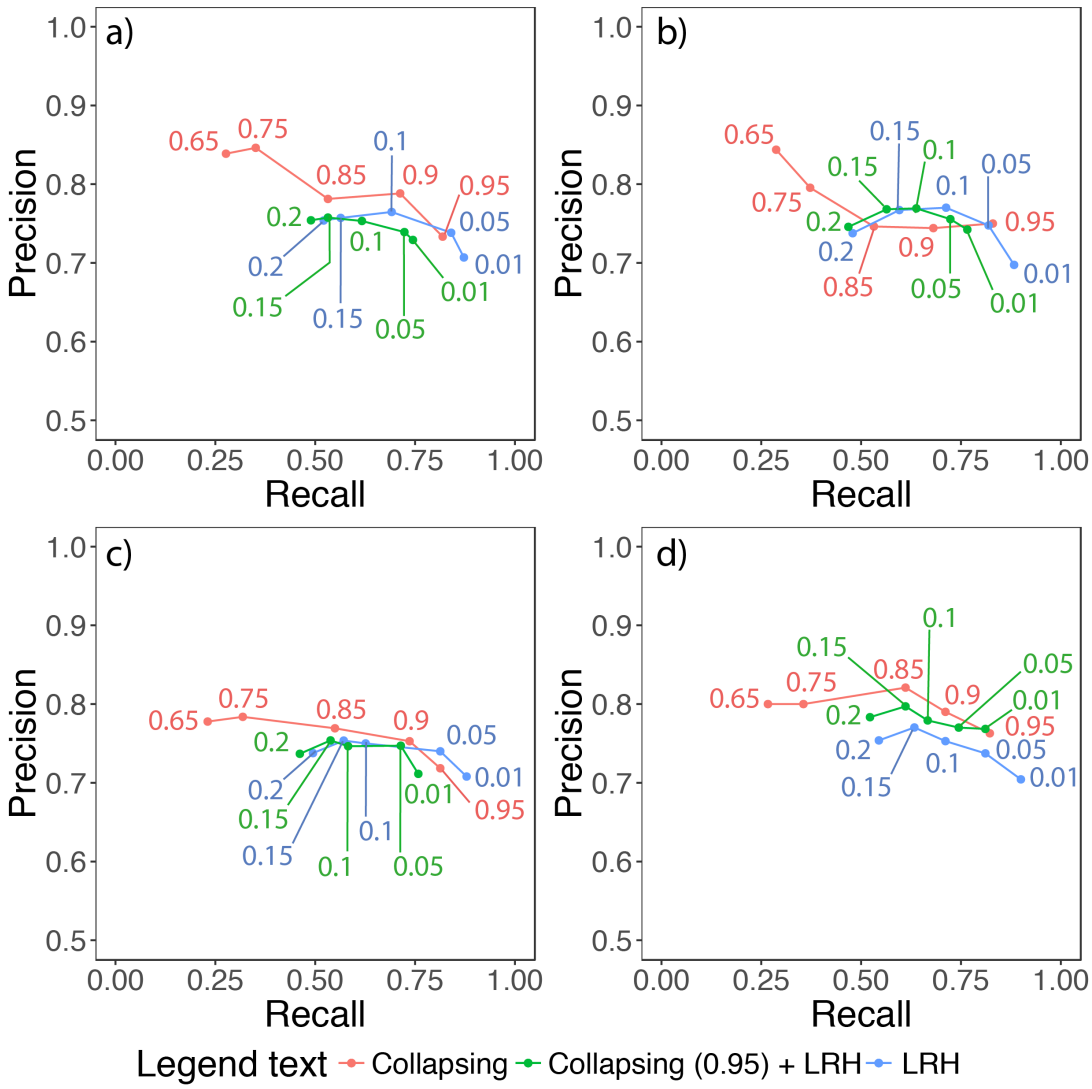


Figure 3.15: Exploring the effect of different RAxML v8.2.12 parameters (Alexandros Stamatakis, 2014): Precision and recall of the tests. Artificially fragmented putative protein sequences constructed on putative high-quality wheat chromosome 3B (Choulet *et al.*, 2014), HOGs with default settings in GETHOGs algorithm (Altenhoff *et al.*, 2013) on the data listed in Table B.1. RAxML parameters: a) `-m PROTCATJTT -c 20`, b) `-m PROTCATLG -c 20`, c) `-m PROTGAMMAAUTO`, d) `-m PROTGAMMAGTR`. Like in the previous analyses, we performed the collapsing approach with a set of thresholds {0.65, 0.75, 0.85, 0.9, 0.95}, the likelihood ratio heuristics (LRH) with a set of significance levels {0.2, 0.15, 0.1, 0.5, 0.01}, and the combination of the collapsing approach (threshold 0.95) and LRH (thresholds {0.2, 0.15, 0.1, 0.5, 0.01}). We again observed consistent outcomes of the tests across trees built with different RAxML parameters. The results were

also consistent with the performance of the tests on trees built with FastTree (Price, Dehal and Arkin, 2010)) (Fig. 3.8b, Fig. 3.13-3.14).

	Recall range			Precision range		
	coll	**LRH**	**comb**	**coll**	**LRH**	**comb**
FastTree default v2.1.7	[0.30, 0.80]	[0.51, 0.92]	[0.43, 0.76]	[0.72, 0.83]	[0.67, 0.78]	[0.72, 0.78]
FastTree double-precision v2.1.10	[0.35, 0.87]	[0.53, 0.92]	[0.51, 0.84]	[0.73, 0.86]	[0.68, 0.78]	[0.72, 0.79]
RAxML v8.2.12	[0.23, 0.83]	[0.48, 0.90]	[0.46, 0.81]	[0.72, 0.85]	[0.70, 0.77]	[0.71, 0.80]

Table 3.3: Brief summary of the results shown in Fig. 3.13-3.15: Recall and precision range for all three tests (collapsing (coll), LRH, combined (comb)) using various settings for two FastTree (Price, Dehal and Arkin, 2010) and one RAxML installation (Alexandros Stamatakis, 2014).

Supplementary files with prediction IDs for all tools and parameters tested can be downloaded from https://doi.org/10.6084/m9.figshare.11733510.v1.

3.3.8 Source code availability

All computer code for heuristics is available for reuse as a user-friendly package ESPRIT 2. It can be downloaded from https://github.com/DessimozLab/esprit2 together with instructions for use. Output files include results of heuristic inference and lists of unambiguous and ambiguous predictions. If an input GFF file is specified, the tool will update it with inferred unambiguous predictions.

3.4 Discussion and outlook

We start this section with a brief overview of the results and the importance of our work. Then we discuss the limitations of the proposed heuristics and outline potential improvements.

3.4.1 Evolutionary inference across species can improve annotation of a target genome assembly

Despite technological and algorithmic advances, genome assembly and annotation remain a challenge, especially for large polyploid genomes with complex evolutionary histories. Putative gene sequences often remain fragmented and fragments get annotated as separate genes. Our work demonstrates that using available assemblies and annotation of related species can provide indications to recognise some of those cases and obtain full-length gene models.

We developed two heuristic approaches and showcase their good performance on a challenging putative proteome of hexaploid bread wheat (*Triticum aestivum* cv. Chinese Spring). In simulations and validation, both of which were performed on the real data taking into account all its complexities, an approach relying on collapsing protein tree branches showed lower recall and higher precision than the likelihood ratio heuristic (Fig. 3.8-3.10). We propose accepting predictions for which both approaches are in agreement, i.e. taking an intersection of predictions obtained by the heuristics as we did in the quest for fragmented gene models in the wheat survey sequence dataset. The performance is even better when we combine the new phylogeny-based combined heuristic with the established pairwise approach ESPRIT (Dessimoz *et al.*, 2011).

Our approaches reveal the power of fine-grained evolutionary inferences across multiple species to improving the quality of genomic data. We hope

they will help make phylogeny-based detection of split gene models a routine step in genome assembly and annotation.

3.4.2 Biological challenges: close paralogs and variable evolutionary rates

The two main inherent challenges of *in silico* split gene model inference are the effect of close paralogs and the variation in the rate of evolution along the sequences.

Indeed, sometimes fragments are derived from closely related or slowly evolving paralogs which code for identical or nearly identical protein sequences, and there is not enough information to distinguish fragments of one gene from another. Hence, we are more likely to make a false positive prediction (Fig. 3.16). This is due to the definitions of our heuristics—they make a prediction if the examined sequences are compatible with the hypothesis that they are derived from the same gene, yet that does not mean the sequences are not derived from paralogs as explained in sections 3.2.2-3.2.3.

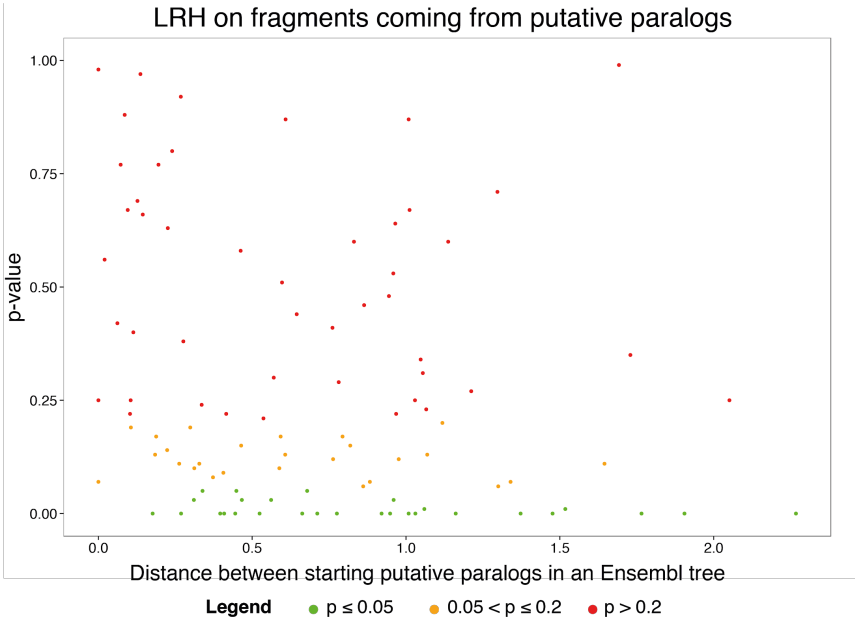


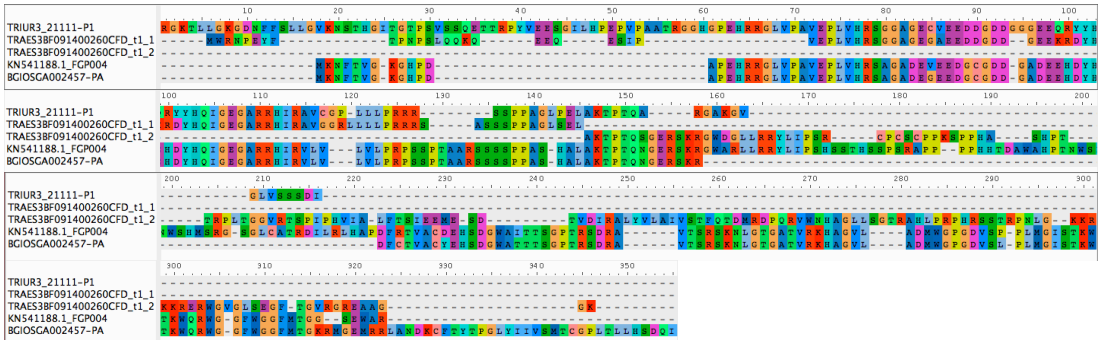
Figure 3.16: The relationship between paralog distance (expected number of changes per site; information exported from Ensembl Plants (Vilella *et al.*, 2009)) and *p*-value for the LRH when applied to random fragments derived from putative paralogs.

As for evolutionary rate heterogeneity across the protein length, this can pose a problem because fragments of the same protein sequence can wrongly appear to be coming from distinct sequences. Consider for instance a protein composed of two domains—one slowly evolving and one fast evolving. If we consider each domain as a distinct sequence and look at their position in an inferred protein tree including full-length putative homologous counterparts, the branch lengths connecting these fragments to the rest of the tree may have markedly different lengths. As a consequence, the increase in likelihood obtained by having distinct branches for each fragment may be sufficiently large for our heuristic to erroneously infer that the fragments come from distinct protein sequences (see example below; Table 3.4 and Fig 3.17). It may be possible to address this problem by more explicitly modelling variation of rate among sites.

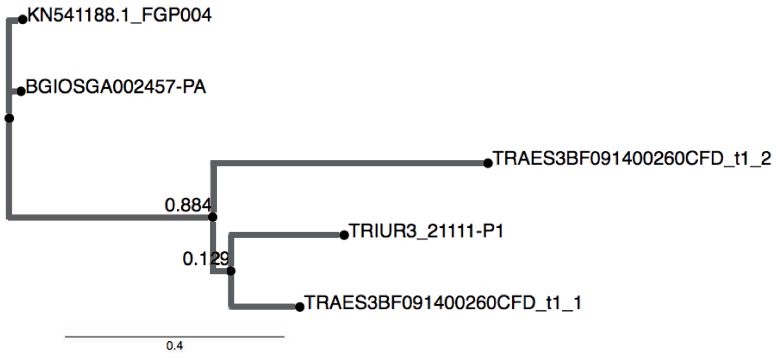
Gene model: TRAES3BF091400260CFD
● *p*-value: 0.02 ● #references in the MSA: 3 ● Length of the MSA: 355 ● Length of the putative protein sequence: 244 ● Length of fragment 1: 92 ● Length of fragment 2: 152 ● Results from the collapsing approach: split gene model
Reference gene model: TRIUR3_21111
● PAM distance (reference, fragment 1): 43.06 ● PAM distance (reference, fragment 2): 734

Table 3.4: Information on a case where the likelihood ratio heuristic does not recognise fragments of the same gene model.
Putative protein sequence TRAES3BF091400260CFD_t1 was split at a random position. Corresponding putative protein family was obtained from Ensembl Plants (Cunningham *et al.*, 2019; Howe *et al.*, 2020), release 31, alignments performed by Mafft v7.164b (Katoh and Standley, 2013) and trees built with FastTree v2.1.8 (Fig. 3.17) (Price, Dehal and Arkin, 2010). The *p*-value of the likelihood ratio heuristic is probably low, i.e. in favour of the hypothesis that fragments are coded by paralogous genes, due to dissimilar evolutionary rates. The collapsing approach correctly infers them as fragments from the same gene when used with any of the thresholds {0.65, 0.75, 0.85, 0.9, 0.95}.

a)



b)



c)

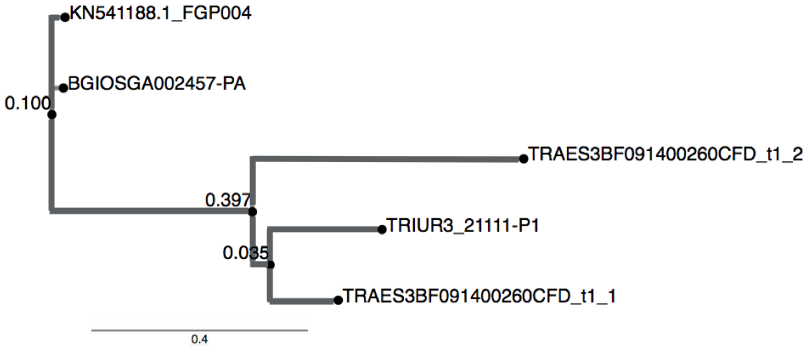


Figure 3.17: Multiple sequence alignment and reconstructed protein tree containing fragments of a putative protein sequence TRAES3BF091400260CFD_t1.

a) Multiple sequence alignment of the putative protein family containing the putative protein (drawn with AliView (Larsson, 2014)), b) Reconstructed protein tree with SH-like branch supports (drawn with Phylo.io (Robinson, Dylus and Dessimoz, 2016)), c) Reconstructed protein tree with branch lengths (drawn with Phylo.io).

Again, the scenario above was to expect given the definitions of our heuristics. The heuristics do not find enough evidence for the sequences being derived from the same gene and automatically accept the hypothesis that they are derived by paralogs as pointed out in sections 3.2.2-3.2.3.

3.4.3 Technical challenges: selection of test parameters and external tools

On a practical level, predictions heavily depend on the choice of two parameters: a threshold for collapsing branches and a significance level for the likelihood ratio heuristics (LRH). Lower, more stringent thresholds for collapsing yield more confident predictions, while higher, less conserved thresholds will produce more, but less confident, predictions. Similarly, a higher significance of the LRH will result in fewer, but more confident, predictions. Obtaining more predictions can be achieved by lowering the significance of the heuristic at the cost of their lower confidence. Overall, it is important to choose thresholds depending on the application. For example, a higher number of predictions can be favourable for comparison with other data.

Predictions also depend on the input putative protein families. Bigger putative families facilitate more predictions (Fig. 3.9a v Fig 3.9b, Fig. 3.9c v Fig 3.9d) but also result in more ambiguous calls, i.e. cases where a fragment is involved in multiple predictions (Tables B.5-B.8). We observed fewer false positive predictions when we simulated fragmentation on bigger putative protein families, where we were more likely to randomly split a pair of putative sequences from more distant paralogs, in comparison to small putative protein families which are more likely to contain only very close putative paralogs (Fig. 3.8a v Fig 3.8b). However, the validation results indicate that the heuristics are still able to identify a reasonable number of split gene models with sufficiently high precision for downstream experimental validation, even when small putative protein families are used (Fig. 3.9). Our attempt to approximate recall values on the low-quality data

confirmed observations on the simulated fragmentation (Fig. 3.8 v Fig. 3.10a-c). More importantly, it also revealed that better and larger putative families could contribute to even better performance of the tests (Fig. 3.10d-f, section B.6). In the four settings analysed, only the minority of potentially fragmented cases (according to the pipeline with BLAST+ mapping) was actually subjected to the heuristics as the fragments were assigned to different putative homologous groups, if assigned at all. So the homology inference algorithm, as well as the putative protein sequences of reference gene models (especially if they are also fragmented), might affect the final predictions even more than we initially thought.

Throughout this project, we fixed some of the parameters. First, we considered only putative protein sequences of gene models at least 50 amino acids long. Shorter putative sequences contain less information thus making phylogeny reconstruction more challenging; at the same time, the benefit of putting together short fragments is also more limited. Second, we required candidate fragments to overlap less than 10%. Increasing the overlap increases the number of candidate pairs and, consequently, the number of predictions including false positive and ambiguous predictions.

Finally, in the experiments described in sections 3.2.7-3.2.9, 3.2.11-3.2.12 we used Mafft v7.164b (Katoh and Standley, 2013) to align putative protein families and FastTree v2.1.8 (Price, Dehal and Arkin, 2010) to reconstruct protein trees, both with their default parameters due to their convenience and speed. Exploring their parameter space or using more suitable tools for the dataset of interest could contribute to higher precision and recall of the heuristics as indicates our examination of FastTree (Price, Dehal and Arkin, 2010) and RAxML (Alexandros Stamatakis, 2014) installations and options (sections 3.2.13 and 3.3.7).

Performing tree reconstruction with different installation versions of FastTree (default v2.1.7, double-precision v2.1.10) (Price, Dehal and Arkin, 2010) and RAxML v8.2.12 (Alexandros Stamatakis, 2014) provided us with both expected and interesting insights. Most importantly, we got consistent results

across the tools and choice of their parameters. Heuristics relying on FastTree double-precision trees were generally slightly more precise and capable to identify fragmentation than the heuristics relying on default installation (Fig. 3.13-3.14, Table 3.3, Tables B.18-B.19). That was indeed what we expected as the double-precision version is recommended for datasets containing nearly-identical sequences (Arkin Lab, 2008) which was the case here. RAxML tree search is more exhaustive than FastTree tree search, hence more likely to find trees closer to the true trees. Thus, we expected an increase in the recall and precision which we observed only in some instances (Fig. 3.13-3.15, Table 3.3, Tables B.18-B.20). Generally speaking, the pipeline employing FastTree double-precision installation outperformed the one with RAxML which we did not anticipate. We acknowledge the small number of cases examined and testing fewer RAxML options than FastTree options mainly due to two reasons: i) author's recommendation not to use his CAT model of rate heterogeneity (A. Stamatakis, 2006; Stamatakis, 2016) on putative families with less than 50 taxa, and ii) computational and time resources needed. However, we included two tree reconstructions with CAT model as well as reconstruction with `-m PROTGAMMAAUTO` option where RAxML uses gamma model of rate heterogeneity (Yang, 1994, 1996) and determines the best substitution model for the data among twenty of them. Further exploration of RAxML parameter space could lead to even better performance of the heuristics while an investigation of predictions on RAxML trees could reveal more stumbling blocks for the heuristics, and perhaps even the ways to overcome these. A larger study including more cases for heuristic inference, cases from different target species, more tree building methods and their settings would be beneficial for more general statements and stronger indications of robustness of the heuristics to the choice of tools and their parameters.

The runtime of the heuristics depends heavily on the runtime of tree reconstruction. Time required for tree reconstruction depends on the dataset, method, number of bootstrap samples for the likelihood ratio heuristic and available computational resources. For example, we encountered putative homologous families for which FastTree (Price, Dehal and Arkin, 2010) built

a maximum-likelihood tree with SH-like branch supports in less than 20 seconds. Trees for different families, as well as trees for bootstrap samples of the same family, can be computed in parallel. Given two tree searches—with and without an input topology—in the best scenario, it takes less than 40 seconds to calculate all the trees in the aforementioned examples. However, we also encountered families for which RAxML (Alexandros Stamatakis, 2014) took 2-3 hours per tree reconstruction. Yet, once the trees are built, the heuristic decision making is performed within single-digit seconds per case.

Some users will have to establish putative protein families in their dataset prior to employing heuristics. That will increase the overall runtime. If the reference proteomes are exported from OMA Browser (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018) and OMA algorithm (GETHOGs) (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2013; Train *et al.*, 2017) is chosen to build putative families, precomputed OMA all-against-all scores can also be downloaded to reduce the amount of computations. Once the target proteome is added to the dataset, reference sequences can be compared only to the target proteome to compute missing pairwise scores and OMA can proceed with homology inference.

3.4.4 Potential improvements

As the large number of detected split gene models in the putative wheat genome illustrates, our heuristic approaches in their present form are already highly useful. However, as often with new approaches, there is still room for improvement.

Currently, when performing the likelihood ratio heuristic, we compute the distribution of the likelihood ratios empirically, via resampling. We computed up to five hundred samples per examined case which, given the simulations and validation, seems to be enough here. Increasing the number of samples might lead to a significantly better approximation of the distribution for cases

where H_s is indeed true, and thus yield more accurate results. In addition, parametrising the distribution of the likelihood ratios would reduce computational time and memory usage. As our preliminary analysis suggests (section B.10), this could be a project itself. Further analyses could also more thoroughly investigate if the range of distribution could indicate putative paralogy and could, perhaps, quantiles indicate risk of misclassification.

We are aware that not all cases subjected to the heuristics are derived from the same gene, which is in contradiction with the assumption of our likelihood ratio heuristic. That partially explains the low precision of the heuristic. A potential improvement could be a heuristic approach consisting of two likelihood ratio heuristics for each candidate pair—one where the hypotheses are as they are now, and one with the reversed hypotheses—which makes inference if both heuristics are in agreement (as suggested for non-nested hypothesis testing in for example Pesaran and Weeks (1999), Lewis, Butler and Gilbert (2011)). An idea is outlined in section B.11.

Since both collapsing and likelihood ratio heuristics rely on evolutionary relationships, some of the mistakes could be avoided by implementing a more realistic evolutionary model. This is of particular importance for cases which are missed due to differences in evolutionary rates across the sequence length.

Some cases of fragmentation are detected but discarded due to ambiguities that arise in other parts of a target gene model (Fig. 3.5c). Looking on a pair-by-pair basis rather than a target gene model-by-target gene model basis, some of the cases could be kept and existing gene models extended. However, due to ambiguities along other regions, the gene model would still remain in the fragmented state (given the evidence at hand).

For datasets with relatively close levels of divergence, using nucleotide instead of amino acid sequences as input might help in correctly distinguishing fragmented gene models from gene models on close paralogs.

It may also be possible to exploit transcriptome data as an additional source of information (Zhang, Zhuo and Hahn, 2016).

To further improve the performance, one could try to find optimal parameters for the dataset of interest and application in question. Different strategies could be used to obtain input putative families, which is one of the key pre-testing steps affecting the predictions, and it might be crucial to consider putative subsequence homology. Alternative tools for alignments and methods with more exhaustive optimal tree search could be also explored. In theory, for a chosen tool or even a set of tools, assuming that they provide trees with comparable likelihoods, the most optimal tree could be chosen among all maximum-likelihood trees reconstructed using multiple starting states (with or without input topology, different input topologies, different random seeds), different substitution matrices, evolutionary models and other tool settings. This would be computationally costly and probably not feasible in practice.

In the current setting, whenever calculating a tree under the H_p, we reconstruct a tree with and without an input topology—roughly doubling the time and computational resources needed for this step. To check whether this could be avoided, we investigated the results of six experiments and present how often a tree search with an input topology yielded more optimal tree, differences between likelihoods (*likelihood of the tree starting from an input topology - likelihood of the tree without an input topology*) and how the more exhaustive search affected the ratio of likelihoods (section B.12). Although brief, the outcomes of the analysis support performing a more exhaustive tree search and including a computation with an input topology. This might not be the case with other tools and their parameter settings.

3.4.5 Recommendations for users

In the current version, we recommend performing both heuristics and taking the intersection of their predictions to increase the reliability of predictions. However, if the goal is to obtain as many plausible pairs as possible, then a union of predictions is more suitable. If due to limited resources only one approach has to be chosen, based on our analyses on wheat, we recommend using the collapsing approach, if higher precision is desired, or the likelihood ratio heuristic (LRH) to obtain more pairs but at a cost of lower precision. In our experiments, a set of predictions obtained by the collapsing approach is a better approximation of the predictions obtained by the combined approach than it is the LRH. Performing LRH might be computationally costly but it can reveal fragmentation unrecognised by the collapsing approach. No matter how small or large the number of predictions unique to LRH is, they can turn out to be of particular interest, especially if they come from a target gene or putative gene family of interest.

In terms of parameters, the higher the collapsing threshold, the more predictions, and the lower the significance level of LRH, the more predictions. However, based on our work so far, the increase in the number of predictions comes at the cost of decreasing precision. If the intersection of predictions is taken, we would recommend running the collapsing approach with the threshold 0.95 and LRH with significance α=0.01. Although these parameters will yield higher number of predictions with lower precision (compared to smaller collapsing thresholds and larger α's), requiring that a split has to be confirmed by both methods will eliminate at least some of the incorrect predictions.

The majority of our applications of LRH was carried out generating 100 bootstrap samples for each examined case. We recommend performing preliminary analyses with 100 bootstrap samples and increasing the number if needed.

3.5 Addendum

The majority of the work presented in this chapter was carried out prior to the release of *Triticum aestivum* cv. Chinese Spring reference genome assembly and its annotation in August 2018 (International Wheat Genome Sequencing Consortium (IWGSC) *et al.*, 2018). Our predictions (section 3.3.6), inferred on an earlier dataset (section 3.2.6) were peer-reviewed and published online in September 2018 (https://doi.org/10.1093/bioinformatics/bty772). Indeed, they could be validated against the reference annotation.

Chapter 4: Detecting fragmented gene models in the putative genome of wild olive, with step-by-step assessments

4.1 Introduction

In Chapter 3 we tackled the problem of fragmentation in annotated genome assemblies by developing new phylogenetic heuristics for identifying potential fragments of the same gene model. A special circumstance of having two versions of assembly and annotation for bread wheat chromosome 3B contributed to the evaluation and better understanding of the performance of the approaches. Yet, typically there is only a single assembly and annotation available, perhaps even produced by the same group of researchers who wish to proceed with downstream analyses which could benefit from improved data quality.

In this chapter, we build upon the work presented in Chapter 3 by using the developed tool ESPRIT 2 (Chapter 3; Piližota *et al.*, 2019) on the only available putative genome of the wild olive at the start of this study (Unver *et al.*, 2017). We provide practical guidance in assessing the method's behaviour on a chosen dataset before embarking on improving the target genome annotation quality. After the method is applied and a set of predictions obtained, we showcase how to assess the results to assure their plausibility. None of the assessments presented here required other sources of data or parameter fine-tuning. This work provides further evidence of the applicability and usefulness of the approach beyond the putative bread wheat genome.

4.2 Methods

The section points out the tool and underlying approaches developed and described in Chapter 3, and the datasets used in this study. Given the absence of a higher-quality genome assembly and annotation of the wild olive genome that could be used as a reference, we assessed the performance of the inference method on artificially fragmented gene models and by manual inspection of predictions on the real unaltered data—the methodological details of which are described here as well.

4.2.1 Dataset

In this study we focused on identifying fragmented gene models in the putative genome of the wild olive tree, *Olea europaea* var. *sylvestris* (Unver *et al.*, 2017) for which 50,684 putative protein sequences were established (more on the data can be found in section 1.3.3). Each of the putative protein sequences was annotated as a product of a different putative gene. Ten plant reference putative proteomes (Table C.1) were exported from OMA Browser, Dec 2018 release (Altenhoff *et al.*, 2018) and putative protein families constructed with GETHOGs algorithm (top-level HOGs) (Altenhoff *et al.*, 2013) with a default set of parameters.

4.2.2 Does it make sense to run the pipeline on the selected dataset?

At the time of writing, this was the only available assembly and annotation for the chosen target species. Thus, to gain insights on the behaviour and performance of ESPRIT 2 (Chapter 3; Piližota *et al.*, 2019) on selected data, we first ran the tool on a dataset comprised of 400 artificially fragmented cases derived from the putative wild olive genome (Unver *et al.*, 2017)—200 cases with two fragments derived from a putative protein sequence of the same gene model and 200 cases derived from putative protein sequences of the putative paralogous gene models. Fragmentation was introduced as

earlier described in Chapter 3, sections 3.2.7-3.2.8. In short, putative protein sequences of randomly chosen gene models (or pairs of gene models in the case of putative paralogs) were cut at a random position so that the resulting non-overlapping fragments were at least 50 AA long.

The inference procedure of ESPRIT 2 (Piližota *et al.*, 2019) combines the two heuristics presented in Chapter 3—the collapsing approach and the likelihood ratio heuristic (LRH), and takes the intersection of their predictions. We applied the heuristics with their default, recommended, parameters: collapsing threshold 0.95 and LRH significance level 0.01 getting an empirical distribution from 100 bootstrap samples (section 3.4.5). Throughout the chapter, we also refer to the inference procedure as “the combined approach”, “the combined heuristic” or “the heuristic”.

Multiple sequence alignments were obtained by Mafft v7.164b (Katoh and Standley, 2013), using its default settings while the phylogenetic trees and SH-like branch supports (Shimodaira and Hasegawa, 1999; Guindon *et al.*, 2010) were computed with FastTree v2.1.7 (Price, Dehal and Arkin, 2010) taking its default set of parameters as well.

After obtaining a set of predictions, we calculated recall and precision for the approach following the same formulas (3.1, 3.2) as in sections 3.2.7-3.2.8:

$$recall = \frac{\#True\ Positive\ predictions\ (TP)}{\#True\ Positive\ predictions\ (TP) + \#False\ Negative\ predictions\ (FN)}$$

$$precision = \frac{\#True\ Positive\ predictions\ (TP)}{\#True\ Positive\ predictions\ (TP) + \#False\ Positive\ predictions\ (FP)}.$$

To supplement information from point estimates, we also modelled recall and precision in a Bayesian framework following the method of Goutte and Gaussier (2005). From a probabilistic point of view, recall can be interpreted as the probability that the heuristic infers fragmentation given that the

examined cases were derived from the same sequence while precision can be defined as the probability that the fragments were derived from the same sequence given that the heuristic inferred fragmentation. In the probabilistic framework, the data (here: true and false positive and negative counts) D = *(TP, FP, FN, TN)* comes from a model with parameters *recall* and *precision* (*r* and *p* from now on) and the values obtained from the formulas above are just estimates of *r* and *p*. Assuming i) the observed counts of *TP, FP, FN, TN* follow a multinomial distribution $D \mid \pi \sim M(n;\pi)$, $\pi \equiv (\pi_{TP}, \pi_{FP}, \pi_{FN}, \pi_{TN})$, $\pi_{TP} + \pi_{FP} + \pi_{FN} + \pi_{TN} = 1$, ii) symmetric beta prior distribution for *r*, i.e. $r \sim Beta(\lambda, \lambda), \lambda > 0,$ and iii) symmetric beta prior distribution for *p*, i.e. $p \sim Beta(\lambda, \lambda), \lambda > 0,$ Goutte and Gaussier showed that posterior distributions for *r* and *p* are also beta distributions:

$$r \mid D \sim Beta(TP + \lambda\ ,\ FN + \lambda\ )$$

$$p \mid D \sim Beta(TP + \lambda\ ,\ FP + \lambda\ ).$$

We chose $\lambda = ½$ and $\lambda = 1$ as they yield commonly used Jeffreys' non-informative prior and the uniform prior respectively (Gelman *et al.*, 2003). For posterior distributions of *r* and *p*, we provide their means (posterior expectation of recall and precision), modes (the most likely values of recall and precision given the data and the model), standard deviations, skewness and 95% credible intervals.

The aforementioned approach involves three assumptions and results with perhaps non-intuitive distributions. For the assumption on multinomial distribution, we first note that our heuristic inference partitions pairs of fragments into four disjoint sets: cases with true positive outcome, cases with false positive outcome, cases with false negative outcome and cases with true negative outcome. Then we interpret the *TP*, *FP*, *FN* and *TN* counts as the result of a multinomial experiment in which elements were randomly and independently drawn from all cases. Second, we assume symmetric beta prior distribution for *r*. Assuming the aforementioned multinomial distribution, it can be shown (Goutte and Gaussier, 2005) that the number of true positive predictions (*TP*) given the number of fragmented sequences (*TP*+*FN*) follows a binomial distribution with parameters *TP* + *FN* and *r*. Beta distributions are

commonly used as a prior distribution for a parameter of a binomial model as they are their conjugate priors which simplifies calculations and interpretation. Furthermore, a symmetric beta prior does not favour neither high nor low recall. Due to the binomial distribution

$$P(D \mid r) \propto r^{TP}(1-r)^{FN}$$

and according to the Bayes' rule

$$P(r \mid D) \propto P(D \mid r)\, P(r).$$

Since $r \sim Beta(\lambda, \lambda), \lambda > 0$,

$$P(r \mid D) \propto r^{TP}(1-r)^{FN} r^{\lambda-1}(1-r)^{\lambda-1}, \text{ i.e.}$$

$$P(r \mid D) \propto r^{TP+\lambda-1}(1-r)^{FN+\lambda-1}.$$

Thus, the posterior distribution of r is $Beta(TP + \lambda, FN + \lambda)$. A similar line of reasoning leads to the third assumption. Again, assuming the specified multinomial distribution for observed counts, it can be shown (Goutte and Gaussier, 2005) that the number of true positive predictions (*TP*) given the number of predictions (*TP+FP*) follows a binomial distribution with parameters $TP + FP$ and p. Following the same reasoning as earlier without favouring high or low values of p, a $Beta(\lambda, \lambda), \lambda > 0$ can be chosen as a prior distribution of p. Similarly as above, since

$$P(D \mid p) \propto p^{TP}(1-p)^{FP} \text{ and } P(p \mid D) \propto P(D \mid p)\, P(p),$$

$$\text{then } p \mid D \sim Beta(TP + \lambda, FP + \lambda).$$

Parameter $\lambda, \lambda > 0$ does not have to be the same for prior distributions of r and p.

4.2.3 Application to the target putative genome

To detect potential fragmentation in the annotation of the selected wild olive putative genome, we used the combined heuristic of the collapsing approach (with collapsing threshold 0.95) and the likelihood ratio heuristic (with significance level 0.01). The pipeline was run in the same way as on the artificial fragments described above in section 4.2.2. We considered only pairs of putative protein sequences (each sequence being at least 50 AA

long) which overlapped for less than 10% of their length in the corresponding multiple sequence alignment.

4.2.4 Assessment of predictions

Being aware that artificial fragmentation does not necessarily capture well fragmentation in the genome annotation and corresponding input putative proteome of interest, and thus, might misguide interpretation of predictions made on the target putative proteome, we also manually inspected 10 randomly chosen predictions on putative wild olive proteome. They can be divided into two groups of 5. The inferred fragments from the first group each belonged to a putative protein family with up to 10 candidate pairs for heuristic inference. The remaining 5 cases were members of more challenging putative protein families—having more than 10 candidates and thus, probably being more fragmented or having higher rates of gene duplication than those in the first group. For each of the 10 cases, we inspected multiple sequence alignment of the corresponding putative protein family and reconstructed protein tree.

4.3 Results

We first present results on artificially introduced fragmentation in a subset of wild olive data. These results encouraged us to apply the method to the whole putative genome and facilitated better understanding of the outcomes. The results of the study are presented in the subsequent subsection. Finally, we describe 10 randomly selected predictions and assess their validity.

4.3.1 Simulated fragmentation in wild olive: recall and precision

To gain better understanding of the behaviour of the combined heuristic on the putative wild olive genome (Unver *et al.*, 2017), we calculated recall and precision on artificially fragmented putative protein sequences of randomly chosen gene models as described above in section 4.2.2.

$$recall = \frac{\#True\ Positive\ predictions\ (TP)}{\#True\ Positive\ predictions\ (TP) + \#False\ Negative\ predictions\ (FN)} = \frac{150}{200} = 0.75$$

$$precision = \frac{\#True\ Positive\ predictions\ (TP)}{\#True\ Positive\ predictions\ (TP) + \#False\ Positive\ predictions\ (FP)} = \frac{150}{204} = 0.735$$

In this particular study, all fragments subjected to the method were at least 50 AA long (Fig. 4.1). Increasing the length threshold and considering only a subset where both fragments were at least 75 or 100 AA long, yielded even higher recall and precision rates (Table 4.1).

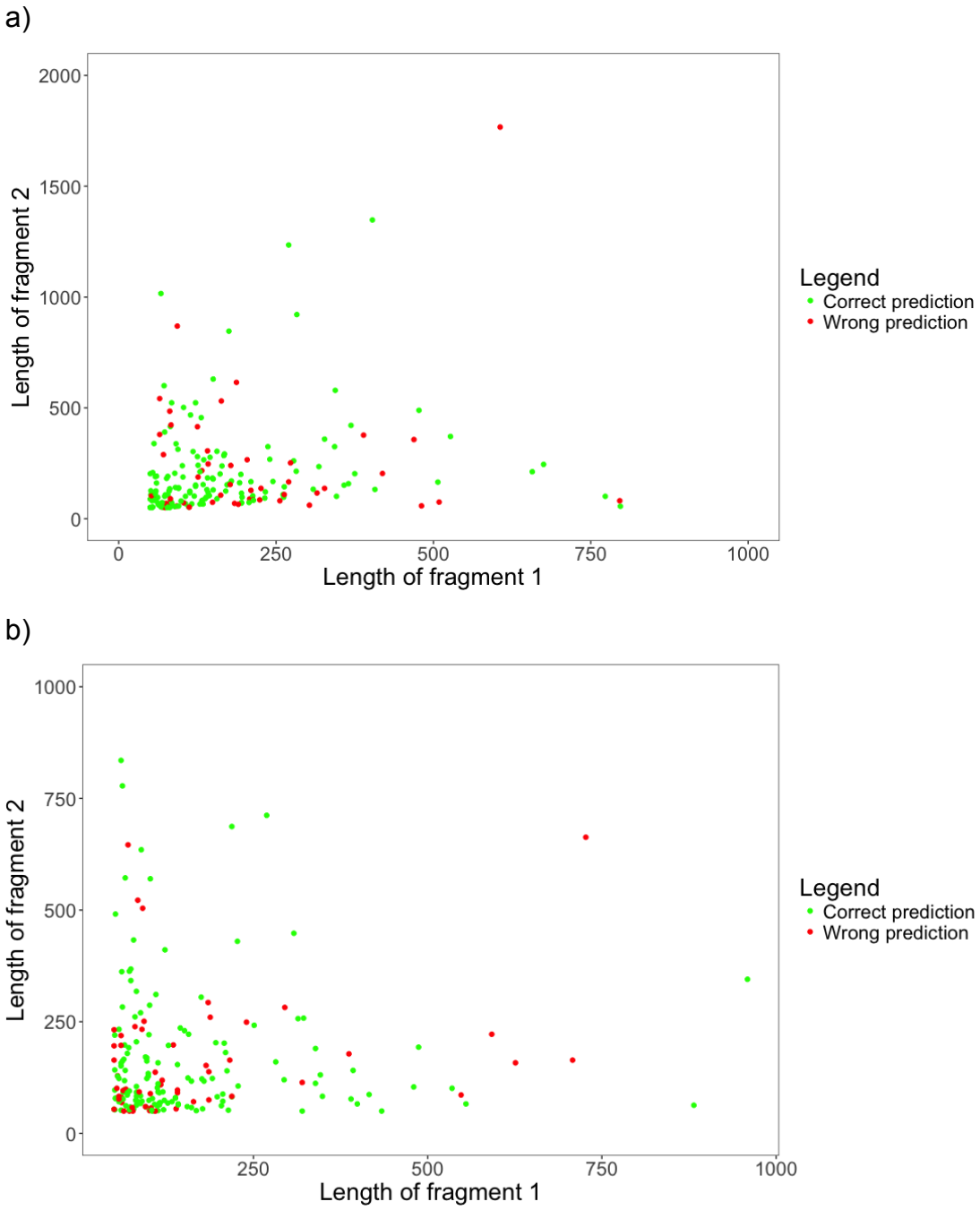


Figure 4.1: Lengths of simulated fragments and outcomes of the combined heuristic.

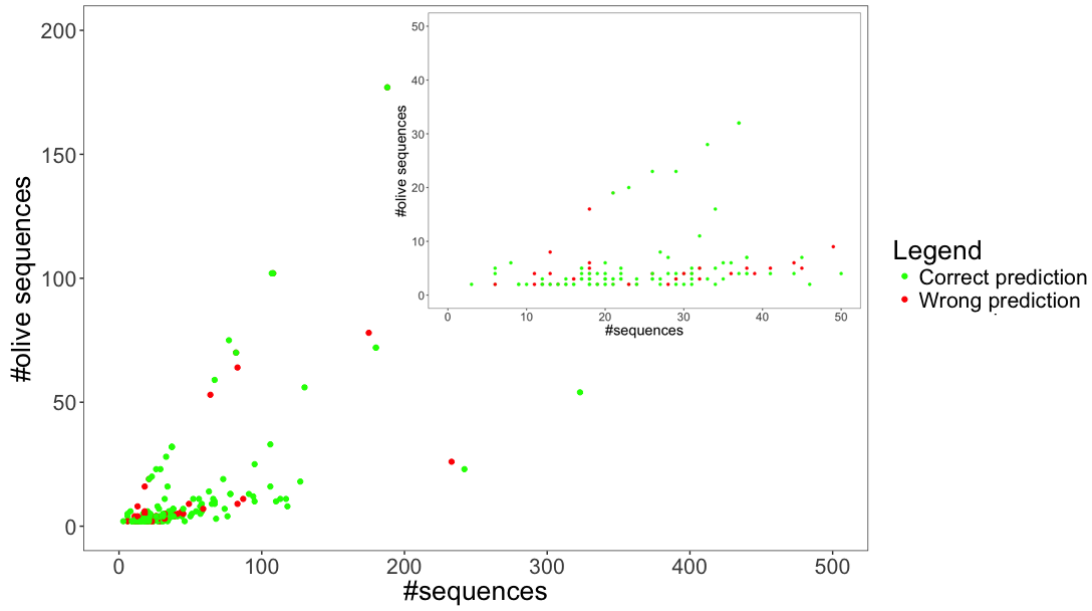
a) Fragments of the same putative sequence, b) Fragments from putative paralogous sequences.

	Recall	Precision
All pairs	0.75	0.74
Pairs where both fragments at least 75 AA long	0.75	0.77
Pairs where both fragments at least 100 AA long	0.76	0.81

Table 4.1: Recall and precision on artificially fragmented putative protein sequences of gene models depending on the length of putative sequences subjected to examination.

For the same set of simulations we also checked trends in recall and precision depending on the numbers of putative sequences (Fig. 4.2). We observed that as the size of putative homologous families increased, recall also increased but at the cost of lower precision (Table 4.2). Using the heuristic on input families with lower proportions of olive sequences, i.e. higher proportion of reference sequences from related species, yielded both higher recall and higher precision (Table 4.3).

a)



b)

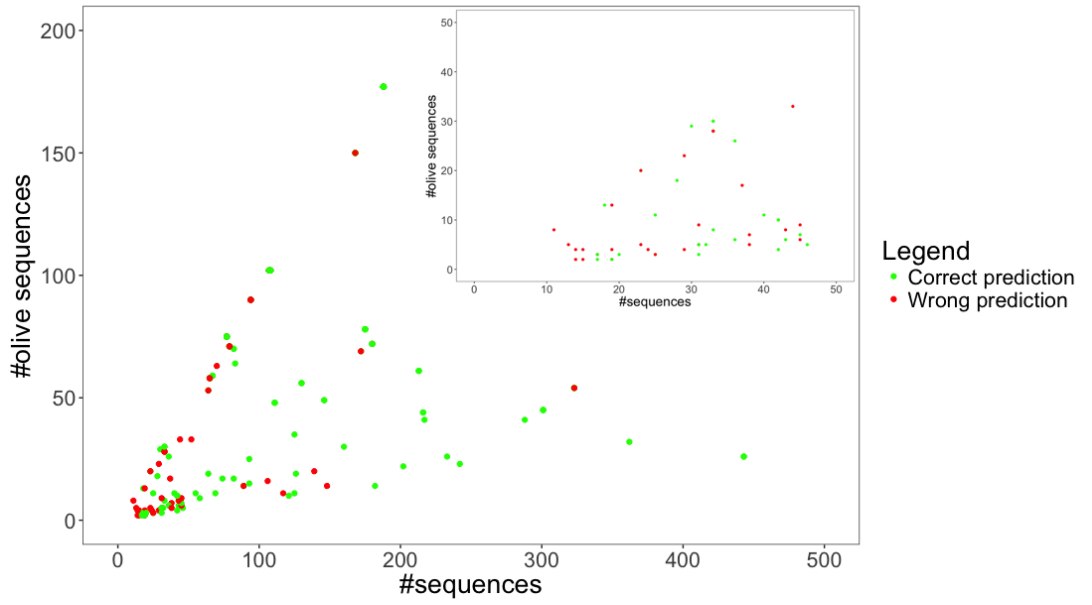


Figure 4.2: Outcomes of the combined heuristic on simulated fragments plotted against the size of putative families (x-axis) and the number of putative olive sequences in the family (y-axis).

a) Fragments of the same putative sequence, b) Fragments from putative paralogous sequences.

	Recall	Precision
All pairs	0.75	0.74
Pairs from putative families with up to 100 members	0.75	0.76
Pairs from putative families with up to 50 members	0.73	0.8
Pairs from putative families with up to 20 members	0.71	0.83

Table 4.2: Recall and precision on artificially fragmented putative protein sequences of gene models depending on the size of input putative protein families.

$\frac{\#wild\ olive\ sequences}{\#sequences\ in\ total}$	Recall	Precision
All pairs	0.75	0.74
≤ 0.75	0.76	0.81
≤ 0.5	0.76	0.83
≤ 0.3	0.77	0.84

Table 4.3: Recall and precision on artificially fragmented putative protein sequences of gene models depending on the fraction of wild olive sequences assigned to a putative protein family.

We also checked outcomes of the combined heuristic on the set of fragments derived from putative paralogs with respect to the percent identity among putative sequences the fragments were derived from. As Figure 4.3 and

Table 4.4 reveal, the mistakes were more frequent among pairs obtained from putative paralogs with higher percent identity. This makes sense, because, as previously pointed out in the context of the wheat genome analysis (section 3.4.2), fragments coded by close paralogs and fragments coded by the same gene become indistinguishable (without additional contextual information).

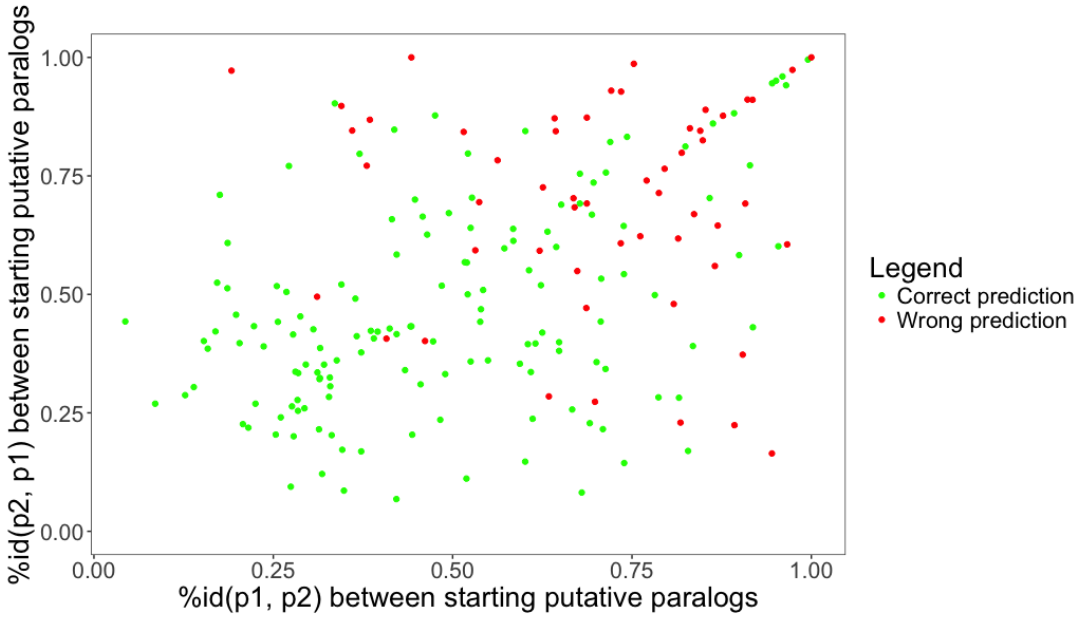


Figure 4.3: Outcomes of the combined heuristic on simulated fragments derived from putative paralogous sequences.
Numbers on the axes represent percent identity among starting putative protein sequences prior to fragmentation.

	Precision
All pairs	0.74
Pairs where both fragments were derived from putative paralogs having protein sequence identity up to 90%	0.79
Pairs where both fragments were derived from putative paralogs having protein sequence identity up to 85%	0.83

Table 4.4: Precision on artificially fragmented putative protein sequences of gene models depending on the percent identity of starting putative paralogous sequences.

In addition, to get a better picture on the variability of recall and precision, we estimated their probability distributions in a Bayesian framework (Table 4.5, Fig. 4.4). We used non-informative prior distributions to “let the data speak for themselves”. Posterior expectations and modes for both recall and precision did not differ much from their point estimates obtained above—recall ~0.75, precision ~0.735. Standard deviation was ~0.03 for all posterior distributions derived. Importantly, 95% of posterior distributions for recall was approximately within interval [0.69, 0.81] and [0.67, 0.79] for precision (more precise numbers in Table 4.5). Small change in the prior distribution parameter (λ) had a small effect on the posterior distribution.

#TP = 150, #FN = 50, #FP = 54	λ =½	λ =1
r ~ Beta(λ , λ)	*Beta(½, ½)*	*Beta(1,1) ~ U(0,1)*
r |D ~ Beta(TP + λ , FN + λ)	*r |D ~ Beta(150.5, 50.5)*	*r |D ~ Beta(151, 51)*
E(r |D)	0.749	0.748
mode(r|D)	0.751	0.75
s(r|D)	0.031	0.030
skewness	-0.161	-0.159
95% credible interval	[0.687, 0.806]	[0.686, 0.805]
p ~ Beta(λ , λ)	*Beta(½, ½)*	*Beta(1,1) ~ U(0,1)*
p |D ~ Beta(TP + λ , FP + λ)	*p |D ~ Beta(150.5, 54.5)*	*p |D ~ Beta(151, 55)*
E(p|D)	0.734	0.733
mode(p|D)	0.736	0.735
s(p|D)	0.031	0.031
skewness	-0.147	-0.146
95% credible interval	[0.672, 0.792]	[0.671, 0.791]

Table 4.5: Summary of derived posterior distributions for recall and precision.

When λ =1, formulas for mode of a beta distribution yield: *mode(r|D)* $= \frac{\#TP}{\#TP+\#FN}$ and *mode(p|D)* $= \frac{\#TP}{\#TP+\#FP}$. Thus, the points in which probability density functions of (unimodal) distributions *r*|D and *p*|D achieve their maximum values correspond to definitions of point estimates for recall and precision (formulas 3.1 and 3.2).

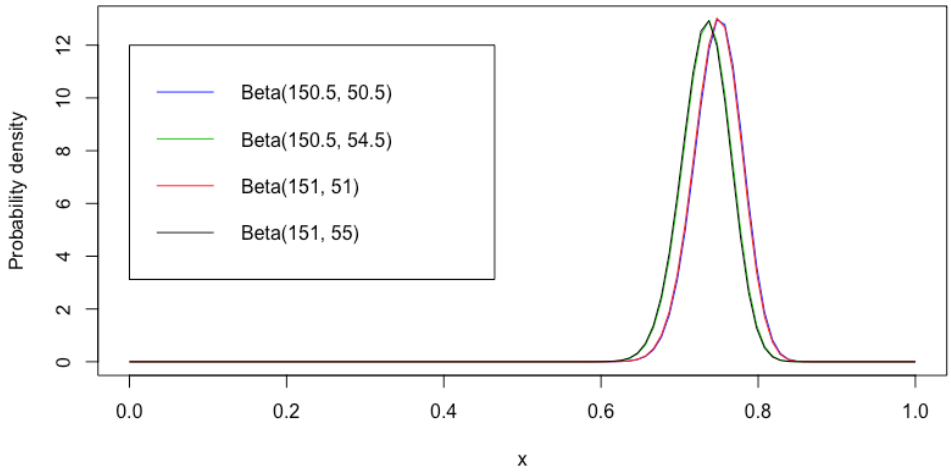


Figure 4.4: Estimated probability density functions from Table 4.5. Recall: *Beta(150.5, 50.5)* and *Beta(151, 51)* (the two "overlapping" on the right), precision: *Beta(150.5, 54.5)* and *Beta(151, 55)* (the two "overlapping" on the left).[46]

Supplementary files with information on the putative protein sequences, corresponding putative homologous families and outcomes of the heuristic inference for all cases can be downloaded from https://doi.org/10.6084/m9.figshare.11702427.v1.

4.3.2 Application to the putative wild olive genome

Phylogenetic heuristics for detecting fragmentation (Chapter 3) were applied to the putative genome of wild olive, *Olea europaea* var. *sylvestris* (Unver *et al.*, 2017), as described in sections 4.2.1 and 4.2.3. In putative protein families containing wild olive and 10 reference plant species (Table C.1), our pipeline identified 2,533 pairs of putative olive sequences that could be subjected to heuristic examination for fragmentation. On closer inspection, we observed that 1,688 candidate pairs were members of a single putative

[46] Source code for the plot (Stephens, 2017) was released under the Creative Commons Attribution International License 4.0 (https://creativecommons.org/licenses/by/4.0). I changed parameters, legend and axis labels in the original code.

protein family having 96 putative wild olive sequences out of 386 putative sequences in total. Due to ambiguities arising from such a large number of combinations of sequences, potentially high level of fragmentation and potentially high rates of gene duplication in the particular family, we proceeded with investigating only the remaining 845 cases spread across other putative protein families.

The phylogenetic heuristics indicated a fair number of fragmented gene models: collapsing with threshold 0.95 found indications that 168 pairs of input gene models were actually parts of the same, longer, model while the likelihood ratio heuristic with significance level 0.01 found indications for 485 pairs. The two heuristics had 166 predictions in common. Some putative sequences were involved in multiple predictions, i.e. the heuristics indicate that some gene models could be fragmented into more than two pieces. If all our inferred fragments spanned different regions of the potential full-length model, we considered corresponding predictions to be unambiguous; otherwise ambiguous (more details in section 3.2.5). Taking into account ambiguities, 166 predictions could be classified into 102 unambiguous and 64 ambiguous predictions.

Although the majority of examined pairs—488 out of 845—belonged to putative protein families with more than 10 candidate pairs (10 families in total), the heuristics found indications for fragmentation in only 2 unambiguous and 20 ambiguous cases (Fig. 4.5). The remaining 357 investigated pairs from putative families containing 10 or fewer cases of potential fragmentation (198 families in total) yielded 100 unambiguous and 44 ambiguous predictions.

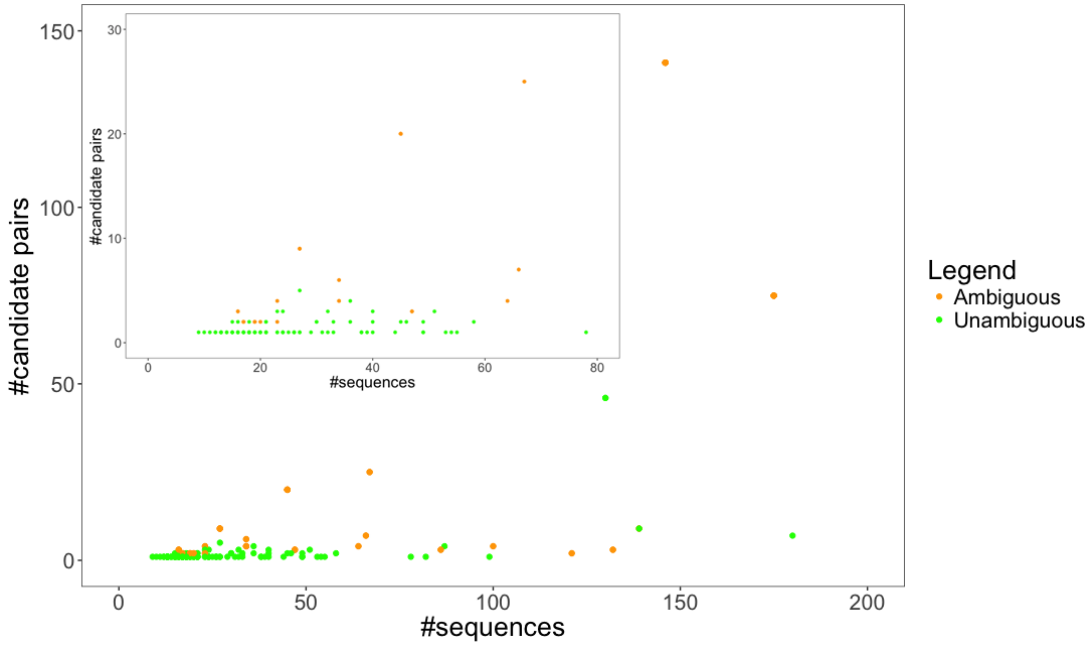


Figure 4.5: Predictions of fragmented gene models with respect to the number of putative protein sequences in the input putative homologous protein family (x-axis) and the number of tested cases in the family (y-axis).

Some points represent multiple predictions. There is only one point on the plot (139,9) which represents both unambiguous and ambiguous predictions (one of each).

For a fixed size of putative protein family, we typically observed unambiguous predictions in families with fewer wild olive putative sequences and ambiguous predictions appearing in those containing more wild olive sequences (Fig. 4.6). We observed a similar trend when we fixed the number of putative wild olive sequences per putative family with respect to the number of pairs subjected to testing—typically the combined heuristic yielded ambiguous predictions as the number of candidate pairs increased (Fig. 4.7).

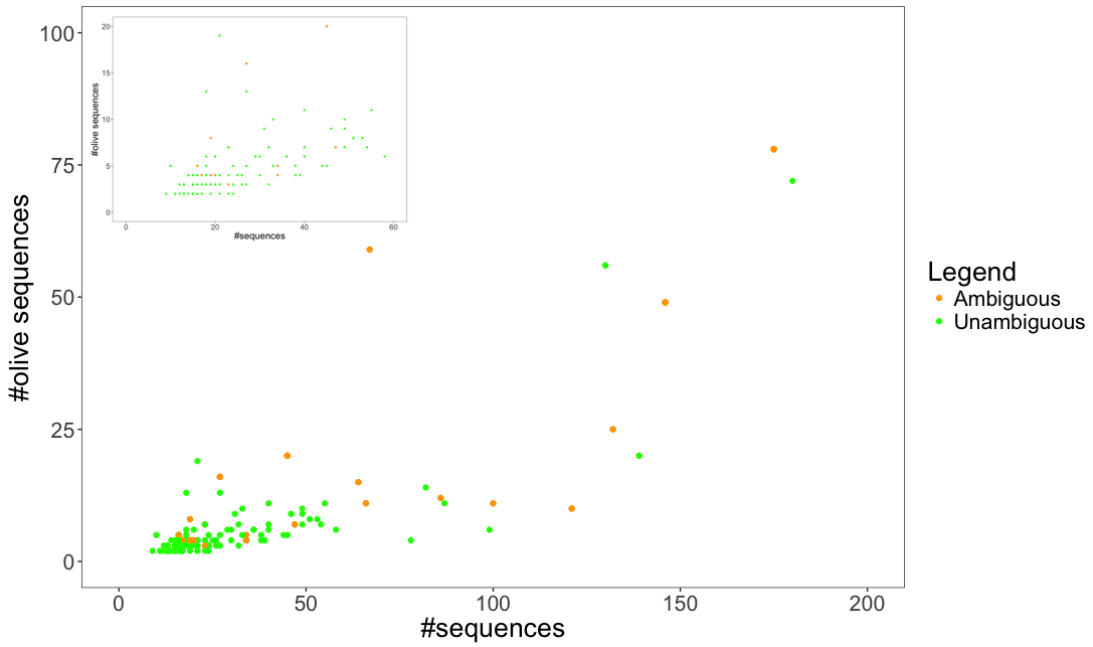


Figure 4.6: Predictions of fragmented gene models with respect to the number of putative protein sequences in the input putative homologous protein family (x-axis) and the number of putative wild olive sequences in the family (y-axis).

Some points represent multiple predictions. Some points represent both unambiguous and ambiguous predictions: (16, 5), (17, 4), (19, 4), (20, 4), (23, 7), (139, 20).

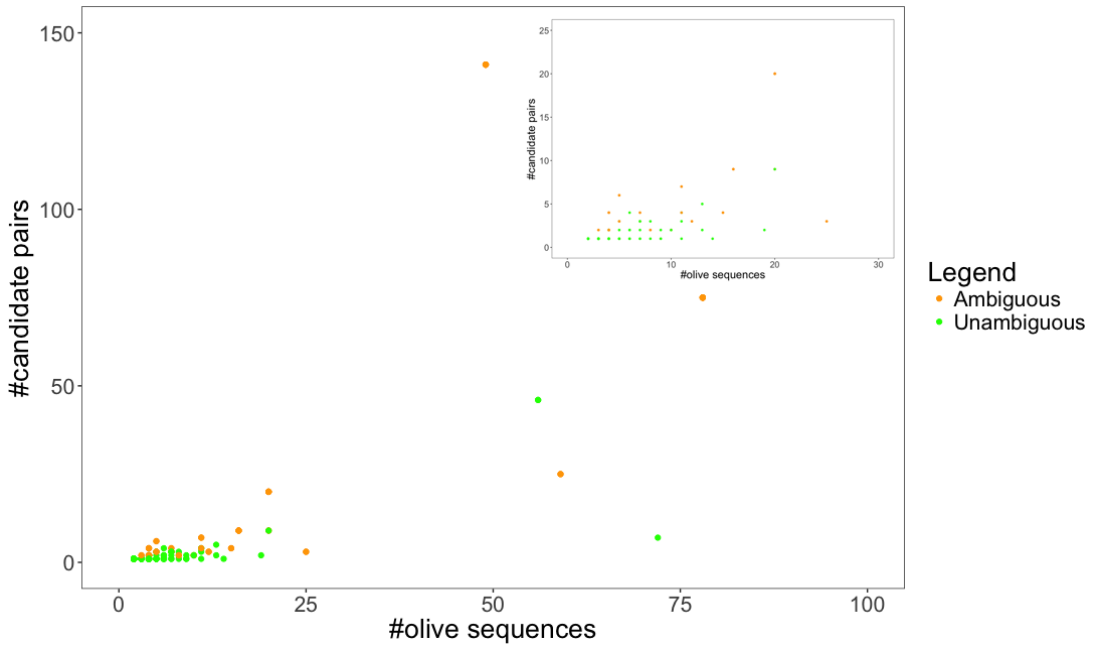


Figure 4.7: Predictions of fragmented gene models with respect to the number of putative wild olive protein sequences in the input putative homologous protein family (x-axis) and the number of cases in the family that were subjected to the heuristic (y-axis).

Some points represent multiple predictions. Some points represent both unambiguous and ambiguous predictions: (4, 2), (5, 3), (7, 3), (10, 2), (11, 4), (20, 9).

Ambiguous and unambiguous predictions were spread evenly with respect to lengths of the corresponding candidate putative protein sequences (Fig. 4.8).

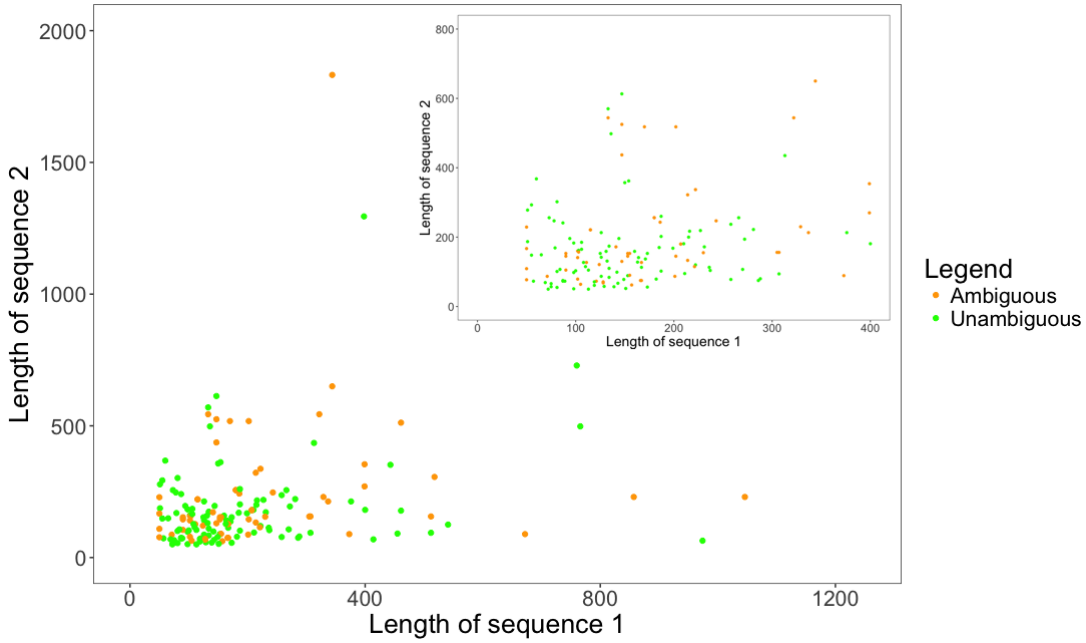


Figure 4.8: Predictions of fragmented gene models in the putative wild olive genome and lengths of fragments of corresponding putative protein sequences.

Supplementary file with details on the predictions and corresponding putative protein families is available at https://doi.org/10.6084/m9.figshare.11702436.v1.

4.3.3 Manual inspection of ten predictions

We manually inspected 10 randomly selected predictions of fragmented gene models in the annotation of the target wild olive genome assembly (section 4.3.2). Five predictions—3 unambiguous and 2 ambiguous—were selected from putative protein families with up to 10 candidate pairs that could be fragments coded by the same gene model (Table 4.6). The other five selected cases—all ambiguous—were in putative protein families with more than 10 candidate pairs (Table 4.7). All three unambiguous predictions seemed plausible while none of the ambiguous predictions seemed

implausible. Based on the information at hand, it was hard to exclude the scenario that the putative sequences involved in ambiguous predictions were not derived from paralogs.

In the rest of the section, we comment on each case, corresponding multiple sequence alignment and reconstructed phylogenetic tree. All alignments and trees can be downloaded from https://doi.org/10.6084/m9.figshare.11702430. Some of them are depicted in Appendix C and referred to in this section. Sometimes we briefly mention other predictions from the same putative family if the family was small or if the putative sequences were placed close to each other in the reconstructed tree of a larger family.

Fragment 1	Fragment 2	Len (fr 1)	Len (fr 2)	#seq in total	#olive seq	$\frac{\#olive\ seq}{\#seq\ in\ total}$	Type of prediction
Oeu002269.1	Oeu041302.1	650	344	17	4	0.24	A
Oeu010989.1	Oeu058850.1	147	437	16	5	0.31	A
Oeu055052.1	Oeu055056.1	237	258	15	3	0.20	U
Oeu003579.1	Oeu046712.1	113	236	33	10	0.30	U
Oeu026692.1	Oeu041610.1	307	94	31	9	0.29	U

Table 4.6: Selected pairs of putative protein sequences assigned to putative protein families with up to 10 candidate pairs for examination. Last column represents the type of prediction–A for ambiguous and U for unambiguous. Sequence lengths in the number of amino acids (Len(fr 1), Len(fr 2)), size of putative protein families (#seq in total) and fraction of reconstructructed wild olive protein sequences in corresponding putative homologous families (#olive seq/#seq in total) go in favour of the reliability of predictions given the heuristic outcomes on simulated fragmentation (Tables 4.1-4.3).

The ambiguous fragmentation prediction (Oeu002269.1, Oeu041302.1) was inferred based on the input putative protein family consisting of 17 putative protein sequences out of which 4 were from wild olive putative proteome. Two pairs were subjected to the combined heuristic approach and the outcomes classified as ambiguous predictions (the other pair being (Oeu041302.1, Oeu035109.1); sequence lengths 344 AA and 1,832 AA). Based on the multiple sequence alignment (MSA) (Fig. C.1) and reconstructed phylogenetic tree (Fig. C.2), both predictions make sense. The

MSA from position 3113 to position 3367 is spanned by sequences Oeu002269.1 and Oeu035109.1 with 96.46% identical residues in the specified region. Thus, the sequences could be derived from paralogs and it is not surprising that the heuristic made two predictions—(Oeu002269.1, Oeu041302.1) and (Oeu041302.1, Oeu035109.1). They could be further experimentally validated.

The ambiguous prediction (Oeu010989.1, Oeu058850.1) was yielded on the input putative protein family containing 16 putative protein sequences out of which 5 were from wild olive putative proteome. Two more pairs from the family, (Oeu010989.1, Oeu017491.1) and (Oeu010989.1, Oeu010988.1) (sequence lengths (147 AA, 525 AA), (147 AA, 130 AA) respectively), were also examined and all three pairs were inferred as potential fragments corresponding to the same gene model—as they all have sequence Oeu010989.1 in common. Based on the multiple sequence alignment (MSA) and reconstructed phylogenetic tree, these predictions make sense. However, the three sequences—Oeu058850.1, Oeu017491.1 and Oeu010988.1— are similar in the MSA region spanning from position 516 to position 638[47] and the heuristic is not capable of making a single unambiguous prediction. Nonetheless, the three ambiguous predictions seem to be good candidates for further examination.

An unambiguous call was made on a pair (Oeu055052.1, Oeu055056.1) assigned to the putative protein family with 15 putative protein sequences, 3 of them being from wild olive. No other pair from the same family was examined. Based on the multiple sequence alignment (Fig. C.3) and reconstructed phylogenetic tree (Fig. C.4), this prediction looks plausible.

Another unambiguous prediction (Oeu003579.1, Oeu046712.1) was made based on the input putative protein family having 33 putative protein sequences out of which 10 were from wild olive putative proteome. One more pair of sequences, (Oeu046712.1, Oeu048697.1), was investigated but

[47] Percent identities in the specified region: (Oeu058850.1, Oeu017491.1) 82.79, (Oeu058850.1, Oeu010988.1) 83.61, (Oeu017491.1, Oeu010988.1) 88.52

not predicted as derived from the same gene. Based on the multiple sequence alignment and reconstructed phylogenetic tree, the prediction looks plausible as well.

The unambiguous prediction (Oeu026692.1, Oeu041610.1) was inferred with the help of information contained in the putative protein family with 31 putative sequences out of which 9 were from wild olive. No other candidate pairs were identified in this family. Based on the multiple sequence alignment and reconstructed phylogenetic tree, this prediction also looks plausible.

Fragment 1	Fragment 2	Len (fr 1)	Len (fr 2)	#seq in total	#olive seq	$\frac{\#olive\ seq}{\#seq\ in\ total}$
Oeu001063.1	Oeu014565.1	167	127	67	59	0.88
Oeu009938.1	Oeu042193.1	145	90	146	49	0.34
Oeu009938.1	Oeu048795.1	145	153	146	49	0.34
Oeu042193.1	Oeu054591.1	90	153	146	49	0.34
Oeu014093.1	Oeu061551.1	399	270	175	78	0.45

Table 4.7: Selected pairs of putative protein sequences assigned to putative homologous protein families with more than 10 candidate pairs for examination.

All of them were predicted as ambiguous. Although bigger families (#seq in total) facilitate higher number of predictions, that can come with higher number of false positive predictions (Table 4.2). The fraction of putative wild olive sequences in corresponding putative protein families (#olive seq/#seq in total) can further challenge inference (Table 4.3). However, given their lengths (Len(fr 1), Len(fr 2) in the number of amino acids), sequence information could aid making true positive and true negative predictions (Table 4.1).

A pair of putative sequences (Oeu001063.1, Oeu014565.1) indicated an ambiguous split gene model with the help of the data in the corresponding putative protein family containing 67 putative sequences out of which 59 were from wild olive putative proteome. The inference was not only challenged by a high fraction of putative olive sequences but with their possible fragmentation as well—25 pairs of olive sequences were identified as candidates for heuristic trial. The selected pair was one of the two that were actually confirmed as a split by the combined heuristic (the other one was (Oeu057720.1, Oeu014565.1) with 111 AA and 127 AA long fragments). Although there were only 8 reference sequences from other species, they were all placed in the vicinity of the two predictions in the reconstructed phylogenetic tree (Fig. C.7). Taking into account multiple sequence alignment (MSA) (Fig. C.5), in particular aligned wild olive putative sequences (Fig. C.6), there are indications of high duplication rate in the wild olive gene family the sequences are derived from. As we can see on the first MSA extract in Figure C.6, the (fragmented) olive sequences tend to span the whole region shown while the coverage is rather sparse on the second extract. This discrepancy in coverage also indicates that there might be some data missing—somewhere along the reads processing, assembling, annotation, putative protein family inference, the data spanning those blank regions was filtered out. To conclude this case, the corresponding MSA and tree, in particular placement of putative sequences from reference species relative to the fragments subjected to investigation, indicate that both predictions make sense but it is hard to further detangle the situation with this data at hand. Given the very challenging setting, the outcome indicating two potential scenarios can direct future attempts in resolving the fragmentation and potentially reduce necessary efforts.

The fragmentation predictions (Oeu009938.1, Oeu042193.1), (Oeu009938.1, Oeu048795.1) and (Oeu042193.1, Oeu054591.1) were derived on the putative protein family with 146 putative sequences, 49 of them being assigned to wild olive proteome. Overall, 141 pairs were trialed yielding 6 ambiguous predictions—the aforementioned three and (Oeu033029.1, Oeu048795.1), (Oeu048795.1, Oeu054591.1), (Oeu033029.1,

Oeu042193.1) with sequence lengths (155 AA, 153 AA), (153 AA, 153 AA) and (155 AA, 90 AA) respectively. All wild olive putative sequences were placed within a larger subtree in the reconstructed phylogenetic tree with a branch leading to the subtree having SH-like support of 1.0. An even smaller subtree (again with a branch leading to it having an SH-like support of 1.0) contains sequences involved in all 6 predictions. Similarly as in the previously analysed case, the tree and multiple sequence alignment indicate possible high gene duplication rate in corresponding wild olive gene family as well as perhaps missing data. Given the data at hand, the predictions do not seem implausible but we cannot claim they are plausible.

The last ambiguous prediction we scrutinised was based on the pair of putative protein sequences (Oeu014093.1, Oeu061551.1). It was found in a putative protein family having 175 putative sequences, 78 of them from putative wild olive proteome. This too was a challenging family, with a total of 75 pairs of putative olive sequences subjected to the heuristic examination. Together with the reconstructed protein tree, the evidence at hand suggests gene duplication in the wild olive. The combined heuristic found indications for fragmentation in 5 more pairs. Similarly as in the previous ambiguous cases, it is hard to tell whether these are true or false positive predictions. Judging by the MSA and computed protein tree, they do not seem implausible.

4.3.4 New ESPRIT 2 output file

To facilitate more convenient exploration of the data and inference outcomes, we extended the output of ESPRIT 2 (Piližota *et al.*, 2019) with a file details_predictions.txt where we provide the following information for each prediction: putative protein fragments' lengths, type of prediction (ambiguous or unambiguous), putative homologous protein family ID, size of the putative family, number of target species putative sequences in the putative family, number of examined pairs in the putative family and number

of predictions in the putative family. This will hopefully shed some light on the data under investigation and help to validate provided predictions.

4.4 Discussion

In this section we provide a brief reminder of the setting of the study, examine behaviour of the prediction method on artificially fragmented putative protein sequences coded by randomly selected gene models, discuss outcomes of the application to the putative genome of wild olive and argue the importance of putting predictions in the context of their corresponding putative protein families.

4.4.1 On the approach

In this study, a set of predictions was obtained using the tool ESPRIT 2 (Chapter 3; Piližota *et al.*, 2019) with its default, recommended parameters (section 3.4.5): 0.95 as a threshold for the collapsing approach and 0.01 as a significance level of the likelihood ratio heuristic. External tools (GETHOGs (Altenhoff *et al.*, 2013), Mafft (Katoh and Standley, 2013), FastTree (Price, Dehal and Arkin, 2010)) were also used with their default settings. No parameter fitting was involved at any step of the pipeline. The input dataset comprised of target and reference species putative proteomes was selected prior to running the pipeline and was not modified later, i.e. no putative proteome was added, removed or altered in any way to tune the results. Taking into account the extensive analyses and validation on the putative wheat genome (Chapter 3), we believe that the outcomes of the study are representative of the tool's performance.

We would still advise a cautious approach to the set of predictions. The following sections discuss ways to examine input putative homologous protein families and potentially fragmented gene models identified by the pipeline.

4.4.2 Simulated fragmentation as a decision-making step

When applied to artificially introduced fragmentation in wild olive putative proteome, the approach yielded moderate recall 0.75 and precision 0.74, both consistent with recall and precision from analogous experiment on wheat (0.72 and 0.73 respectively, sections 3.3.1 and 3.3.2). Further look into the input data and outcomes of the heuristic trial did not reveal unexpected stumbling blocks for heuristic inference. The approach yielded higher recall and higher precision as the lengths of fragments increased—expected given that longer putative sequences are more likely to contain more information which then aids inference. With the increase of the size of input putative homologous families, the recall increased at the cost of lower precision. On one hand, more sequences can potentially provide more useful information but on the other, they might confound the inference, especially if they are fragmented. Indeed, when we inspected behaviour of recall and precision based on the proportion of putative wild olive sequences in a putative homologous family, both recall and precision increased as the fraction of olive sequences decreased. Finally, we checked false positive predictions for fragments derived from putative paralogs and observed that the frequency of mistakes increased with the increase of percent identity among corresponding starting unaltered putative sequences.

Given the observed outcomes of the inference method, we also estimated (posterior) probability distributions for recall and precision. For both recall and precision, the mean and mode of the corresponding distributions were less than 0.01 away from- or equal to the earlier discussed point estimates, standard deviation was 0.03 and 95% credible intervals were roughly between 0.67 and 0.81. Assuming the sample was a good representative for real fragmentation in the wild olive putative genome and the Bayesian modelling of recall and precision was based on reasonable assumptions, these results show great promise.

Overall, the experiments on simulated splits yielded the anticipated behaviour of the combined heuristic approach. Although fragmentation was

artificially introduced, the outcomes suggest that application of the pipeline to the whole wild olive putative genome is a reasonable attempt in identifying fragments of the same gene models.

4.4.3 Application to the putative wild olive genome: detecting and scrutinising detected fragmentation

The aim of the study was identification of fragments of the same gene model in the putative genome of the wild olive tree, *Olea europaea* var. *sylvestris* (Unver *et al.*, 2017).

We applied our heuristic approach to 845 cases and found indications for fragmentation in 166 of them (102 unambiguous, 64 ambiguous). Although the majority of cases (488/845) was assigned to putative protein families where more than 10 candidate pairs were trialed, they yielded only the minority of predictions (2 unambiguous, 20 ambiguous). This indicates that the number of predictions does not merely depend on the number of candidate cases but also on the information contained in corresponding putative protein families. In fact, a rather low number of fragmentation predictions, but reliable predictions, is favourable if the family is highly duplicated or fragmented—both challenges present in the olive dataset and likely in these putative families. For a fixed size of putative homologous protein family, we typically observed that unambiguous predictions occurred in the families with fewer putative wild olive sequences and ambiguous predictions in the families with more wild olive sequences. Similarly, by fixing the number of putative wild olive sequences per putative family, we observed that unambiguous predictions were obtained when the number of candidates was lower and as it increased, ambiguous predictions appeared. Both observations are anticipated and support the idea of getting reasonable outcomes by running the pipeline on this particular dataset.

We manually inspected 10 randomly selected cases and none of them seemed implausible (section 4.3.3). More precisely, 3 unambiguous

predictions looked plausible while the remaining 7 ambiguous cases could be coded by paralogous genes as well as by the same gene. Five cases (3 unambiguous, 2 ambiguous) were chosen from putative homologous protein families with no more than 10 pairs subjected to heuristic examination of fragmentation. The remaining five ambiguous cases were selected from putative families with more than 10 tested pairs. Former cases belonged to smaller and more conserved putative families which led to better resolved reconstructed phylogenetic trees (higher internal SH-like branch support values). In inspected reconstructed phylogenetic trees putative olive protein sequences were usually placed within olive-only subtrees, i.e. subtrees containing no putative sequences from other species. Furthermore, usually the branch attaching the subtree to the rest of the tree had high SH-like support, higher than 0.95. When this branch had support higher than the collapsing threshold (here 0.95), only cases within the subtree could potentially be identified as fragments which causes missing out predictions if one of the fragments is outside the subtree. Potential polyploidization events that led to the extant wild olive genome and estimated high heterozygosity rate (section 1.3.3) also pose biological challenges for the test. Yet, taking into account the list of predictions and having in mind that there were not many predictions on families with >10 candidate cases (section 4.3.2), we have reasons to believe that our heuristic approach was capable of dealing with these issues on this particular dataset with external tools and corresponding settings used along the pipeline.

We encountered a putative protein family containing 386 reconstructed sequences out of which 96 were derived from wild olive forming 1,688 candidate pairs for heuristic examination. The matter was further complicated with 108 *Glycine max* and 160 *Zea mays* sequences (<10 putative sequences per other reference species). As we had not done any approach validation on the data with that extent of fragmentation and potentially high rates of gene duplication and heterozygosity, we refrained from making any predictions. Although our approaches attempt to resolve challenging cases which remained fragmented despite extensive assembling and annotation efforts prior to genome release (and often accompanying peer-reviewed

paper), we thought that this putative homologous family could be too challenging to reliably disentangle 1688 cases.

4.4.4 Final remarks on quantifications

Throughout this chapter, especially in sections 4.3 and 4.4, we extensively classified and quantified the input data, predictions, types of predictions. These numbers can be a handy sanity check and immediately direct users to the peculiarities of the dataset or dubious performance of the fragmentation detection approach(es). We encourage users to put their predictions in the context of putative protein family size, number of target species putative protein sequences within a putative family, number of candidates for heuristic examination within a putative family and length of putative protein sequence fragments. It might be possible to filter out some candidates if the numbers suggest a very challenging scenario, e.g. not many reference putative sequences or too many potentially fragmented candidate pairs within a single family. For already inferred predictions, these numbers can help deciding which ones could be more or less reliable—the more challenging the scenario, the less reliable the predictions[48]. We hope that a new output file of our tool ESPRIT 2 (Chapter 3; Piližota *et al.*, 2019) will be useful in that regard. An even better picture can be obtained if multiple sequence alignment and reconstructed protein tree are also considered but we are aware that requires substantial manual work and may only be practical for a small number of cases.

Creating a universal framework for validation of predictions is a complex task when identifying fragmentation in annotation of *de novo* assemblies which represent the best attempt in reconstructing gene space of species under investigation. We do not know the truth and we try to infer it with our limited knowledge. We do know some factors that affect inference, yet it is hard to quantify their effect, especially as they might vary from case to case. For

[48] This is a generalisation which may not hold true for particular cases. Also, a user might make misjudgements on the situation under investigation.

example, consider assigning a confidence score to each prediction. Ideally, it would reflect the evolutionary history (relationships and distances among species in the dataset, species-specific evolutionary events, rates of evolution), quality of the data (sequencing, assembling, annotation, putative gene/protein family inference, multiple sequence alignments, phylogenetic tree reconstruction) and the capability of our approach to deal with them. It is reasonable to assume that the higher the data quality, the higher the probability of correct inference, yet the question is how to quantify the data quality, its impact on the outcome of our approaches and its interplay with other factors. Does the whole genome duplication within the target species introduce more challenges than fragmentation within a reference species? Can variable evolutionary rates along the target gene sequence get recognised with enough high-quality data from reference species? When and to what extent? How do external tools in the pipeline affect predictions? We saw in Chapter 3 that many pairs of potential fragments could not be found in the same putative protein family (sections 3.3.3 and B.6). How do the heuristics behave on a particular dataset, on a particular test case? For example, from the wheat study it does not seem that empirical distribution of the likelihood ratio heuristic can be easily generalised (sections 3.4.4 and B.10). Thus, relying on a researcher's expertise in the species within a dataset, especially in target species, and its ability to judge based on the observed properties of the dataset and prediction outcomes, might be the most pragmatic approach to assure that predictions are plausible, to get a sense of prediction confidence and to make decisions on further, more extensive and laborious, validation.

Chapter 5: Identifying fragments of the same transcript model in transcriptome datasets, with putative cassava transcriptome as a test case

5.1 Introduction

As already mentioned in Chapter 1, one of the common features of genome and transcriptome assemblies is that both often suffer from fragmentation. A certain degree of it arises from the incapability of the current scientific methods to deal with the same biological features present in both transcriptome and genome data, e.g. gene duplication, ploidy, repetitive regions. However, there are also differences between the two. Typically the aim of transcriptome assembly is to obtain representation of expressed transcripts in a sample rather than obtaining one continuous sequence representation per chromosome like in genome assembly. Yet, transcriptome assembly is not an easy task. It deals with reconstructing different types of RNA molecules, pervasively transcribed regions, alternatively spliced transcripts of the same gene and transcripts expressed at low levels which could even be discarded as assembling artifacts if assembled at all. Hence, methods developed for genome assemblies can be a good starting point for transcriptome assemblies but will require certain modifications to deal with transcriptome data.

Transcriptome assemblies are not spared from misannotations analogous to the ones observed in genome assemblies. In particular, fragments coming from the same transcript are often annotated as multiple separate transcripts from a single or multiple genes. In this type of data, as well, some of the fragmented models could be detected with the help of (putative) homologous genes from related species.

In this chapter, we assemble and annotate a putative transcriptome of cassava, *Manihot esculenta* cv. 98/0581, and explore the ability of previously developed phylogenetic heuristics (described in Chapter 3) and ESPRIT (Dessimoz *et al.*, 2011) to detect non-overlapping or slightly overlapping fragments of the same transcript model. Then we validate our predictions against publicly available reference putative cassava proteome and discuss their nature. The annotation of input *de novo* transcriptome assembly likely contains fragmented transcript models while the high-quality reference proteome facilitates the classification of predictions into true positives and false positives. Given the limited time frame of the project, the results are not yet conclusive but we hope that our suggestions for future work will aid gaining better understanding of the underlying causes and provide insights for future developments.

5.2 Methods

In this section, we briefly summarize heuristics for detecting fragmented gene models that were used in the study, explain the process of obtaining the input data and describe validation of acquired predictions.

5.2.1 Approaches for detecting fragmentation

To infer fragmented transcript models, we used two phylogenetic heuristics developed and described in Chapter 3, and an established software package ESPRIT (Dessimoz *et al.*, 2011). The first phylogenetic approach collapses low-support internal branches of a reconstructed phylogenetic tree with the expectation that putative protein-coding sequences of fragments derived from the same transcript will become sister leaves after collapsing. In the second approach, we performed a likelihood ratio heuristic (LRH) with the null hypothesis being that the two models under examination come from the same transcript. We also investigated predictions of a complementary tool ESPRIT which exploits putative pairwise relationships among putative

proteins from related species. Furthermore, we performed a meta-approach ESPRIT+LRH where we took a union of predictions obtained by ESPRIT and the LRH.

5.2.2 Cassava: from raw reads to coding regions within transcript models

As a test case for the study, we used a putative cassava transcriptome which we assembled *de novo* from raw sequencing reads and then annotated. On such a transcriptome assembly which is not further improved by additional sources of data or manual curation, we expect the presence of fragmented putative (protein-coding) transcripts—putative transcripts that could be identified by our heuristics.

We downloaded Illumina RNA-Seq paired-end reads from NCBI (NCBI Resource Coordinators, 2017). According to the sample description, the RNA samples had been collected from leaves of 7-week-old *Manihot esculenta* cv. 98/0581, purified using Illumina Ribo-Zero magnetic kit[49], prepared for sequencing using Illumina TruSeq Stranded mRNA preparation kit with poly(A)+ selection protocol, and then sequenced on Illumina HiSeq 2500 machine. Sequencing reads had been deposited in NCBI's (NCBI Resource Coordinators, 2017) public repository under the BioProject PRJNA282938 (ARC-OVI, 2015). The dataset, SRR4019554_1.fastq and SRR4019554_2.fastq, contained 12,509,296 pairs of raw reads.

Preprocessing and cleaning of the reads was performed according to the tutorial on best practices for transcriptome assembly with Trinity (Grabherr *et al.*, 2011; Freedman, 2016) depicted in Figure 5.1. First, we examined quality metrics for the raw reads using FastQC (Andrews, 2010). Then we cleaned the data in four steps. Erroneous *k*-mers which could impact assembly were identified and removed with Rcorrector (Song and Florea, 2015), low complexity reads and reads containing significant number of unknown

[49] Removes cytoplasmic, mitochondrial and chloroplast rRNA (Illumina, Inc., 2020).

residues were removed with a Python script provided in the tutorial, and Trim Galore! (Krueger, no date) was used to remove adapter contamination and low quality bases in reads. Finally, despite using poly(A)+ selection in the library preparation for sequencing, some reads in the dataset might be coming from rRNA. To identify and remove them, we mapped the data against the SILVA database of rRNA reads (Glöckner *et al.*, 2017) using Bowtie 2 (Langmead and Salzberg, 2012). Reads which did not align to the rRNA database were checked for quality, again using FastQC and used as input for Trinity. All tool parameters were kept the same as in the tutorial, except for the time and memory requirements which depend on the size of the dataset and available resources on the computer clusters employed for computations. Stranding protocol in RNA-seq library preparation kit was also taken into account.

To assess the quality of the assembly, we checked basic assembly metrics with TransRate (Smith-Unna *et al.*, 2016), quantified read support, i.e. the percentage of reads where both ends align to the same contig, using Bowtie 2 (Langmead and Salzberg, 2012), and finally, assessed completeness of the assembly with BUSCO (Simão *et al.*, 2015) using their dataset of putative near-universal single-copy orthologous genes in plants.

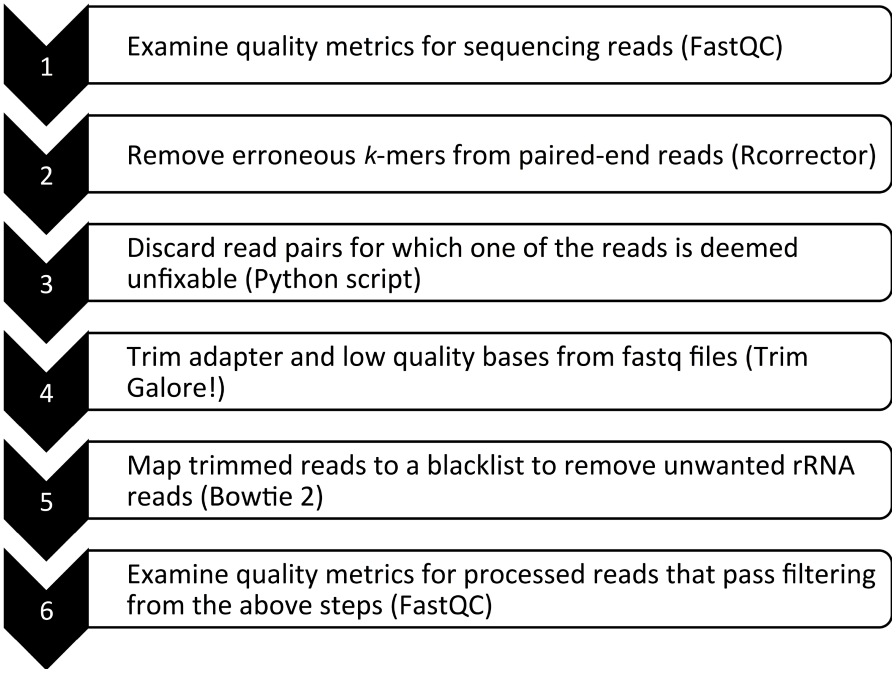


Figure 5.1: Preprocessing and data cleaning procedure as suggested by Freedman (2016).

Finally, since all approaches require putative proteome input data, we predicted and translated protein-coding regions on the contigs obtained by Trinity (Grabherr *et al.*, 2011). We used TransDecoder (Haas and Papanicolaou, 2016) and included homology search with Pfam (Finn *et al.*, 2014) as open reading frame retention criteria. Completeness of the putative gene set was again assessed with BUSCO's plants reference set (Simão *et al.*, 2015).

5.2.3 Improving cassava transcriptome: identifying fragments from the same gene transcript

All three approaches—the two phylogenetic heuristics and ESPRIT (Dessimoz *et al.*, 2011)—were applied on a dataset comprised of predicted cassava peptide sequences (see 5.2.2 above) and five reference plant proteomes downloaded from OMA Browser (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018), March 2017 release (Table D.1). We also considered a union of predictions from ESPRIT and the likelihood ratio heuristic with a significance level of 0.01.

To get candidate fragments and corresponding putative protein families for the heuristics, we followed the procedure outlined in Chapter 3. To obtain putative protein families, we ran the 'bottom-up' variant of GETHOGs algorithm (Train *et al.*, 2017) with default parameters, and used the inferred root-level Hierarchical Orthologous Groups (HOGs) as input for the heuristics. In this process, the GETHOGs algorithm retains only one putative isoform sequence per putative gene—the one with the highest number of significant matches in other species during the all-against-all step in the homology inference, thus showing the potential to have the highest number of inferred evolutionary relationships across species in the dataset (Altenhoff *et al.*, 2010). For the candidate pairs, we considered putative sequences which were each at least 50 AA long and which mutually overlapped less than 10% in the multiple sequence alignment of the corresponding putative protein family. Again, we used Mafft v7.164b (Katoh and Standley, 2013) to align putative protein families and FastTree v2.1.8 (Price, Dehal and Arkin, 2010) to reconstruct protein trees and obtain SH-like branch supports (Shimodaira and Hasegawa, 1999; Guindon *et al.*, 2010). We ran both tools with their default sets of parameters.

We applied three methods for detecting split gene models: i) we collapsed tree branches with SH-like support less than 0.95; ii) performed the likelihood ratio heuristic (LRH) with significance levels of 0.2, 0.15, 0.1, 0.05 and 0.01 using 100 bootstrap samples, and iii) ran ESPRIT (Dessimoz *et al.*, 2011) with default parameters. Note that ESPRIT does not try to discard any putative protein isoforms and treats them all as independent models. This can potentially translate into many ambiguous predictions. As already pointed out in section 3.2.5, some fragments might be involved in multiple predictions made by the same heuristic. For such predictions, we inspect corresponding multiple sequence alignments and resolve ambiguities where possible. More precisely, when fragments from conflicting predictions span different regions of the alignment, we unambiguously accept all predictions. ESPRIT does not try to resolve ambiguous predictions and all pairs having a fragment in common are considered ambiguous. Finally, we also considered

an approach which takes advantage of both pairwise and phylogenetic settings. More precisely, we took union of predictions inferred by ESPRIT and the LRH with a significance level of 0.01.

5.2.4 Validation of predictions

To assess the performance of the heuristics, we validated predictions against the putative high-quality reference proteome of cassava, *Manihot esculenta* cv. AM560-2 (Bredeson *et al.*, 2016), using the same approach as in Chapter 3 (described in 3.2.9 and B.3.1). The putative reference genome (assembly v6.0, annotation v6.1, downloaded from Phytozome (Goodstein, D. M. *et al.*, 2012; Phytozome, 2017)) contained 33,033 annotated protein-coding genes—more than 99% of the estimated number of genes (more on the data quality and completeness in section 1.2.4), and we used the corresponding putative proteome containing one putative protein sequence per putative gene. All putative sequences involved in predictions obtained by heuristics and all sequences involved in ESPRIT's (Dessimoz *et al.*, 2011) unambiguous predictions were queried against the putative reference proteome using BLAST+ v2.2.30 (Camacho *et al.*, 2009). Accounting for differences between our transcriptome and the reference proteome arising from sequencing data and different assembly and annotation pipelines for different cultivars, we required a query coverage of at least 95% having at least 95% identical residuals in the matching regions of a query and the corresponding hit.

5.3 Results

5.3.1 Reasonably good cassava transcriptome assembly

As already detailed in Methods (see section 5.2.2), Illumina paired-end reads were first subjected to quality assessment, cleaning and then again quality assessment prior to assembling. This reduced the initial set of 12,509,296 pairs of raw reads to the set of 10,679,994 pairs of reads. All starting reads were 100 bp long while after processing lengths varied from 36 to 100 bp. FastQC (Andrews, 2010) analysed per base sequence quality, per sequence quality scores, per base sequence content, per sequence GC content, per base N content, sequence length distribution, sequence duplication levels, overrepresented sequences, adapter content and *k*-mer content of the reads. In the assessment of the final set of reads, it reported "FAIL" status only for "Per base sequence content" and "Kmer content" which was expected and acceptable. More precisely, due to the bias in the random selection of sequencing primers, nearly all RNA-Seq libraries fail the "Per base sequence content" module (Babraham Institute Bioinformatics Group, no date). The problem cannot be fixed by processing but importantly, biased choice of primers does not mean individually biased sequencing reads. Furthermore, such a library will almost always fail the "Kmer content" module (Babraham Institute Bioinformatics Group, no date).

The final set of processed reads was assembled *de novo* using Trinity (Grabherr *et al.*, 2011). The resulting 80,917 contigs, including all putative isoform sequences recognised by Trinity, had N50 of 1,799 bp. The shortest contig was 201 bp long (Trinity retains only contigs ≥ 200 bp) and the longest 15,670 bp with mean length of 1211.38264 bp. Almost half of the contigs (39,138; ~48.37%) were longer than 1,000 bp and 17 contigs were longer than 10,000 bp. Considering only the longest putative isoform sequence per (Trinity) 'gene' yielded N50 of 1,702 bp. Out of all paired-end reads provided as input for Trinity, 66.56% could be mapped to the resulting contigs so that

both ends align to the same contig. BUSCO (Simão *et al.*, 2015) identified 1,231 out of 1,440 (85.5%) models of complete putative near-universal single-copy orthologous genes, while further 72 gene models (5%) were classified as fragmented and 137 models (9.5%) could not be found in the assembly. Despite performing only *de novo* assembling, the resulting assembly was fairly complete with regard to annotated putative near-universal single-copy orthologous genes.

Trinity (Grabherr *et al.*, 2011) contigs were used as input for TransDecoder (Haas and Papanicolaou, 2016) which predicted 57,252 protein-coding regions including isoforms. Protein-level BUSCO assessment (Simão *et al.*, 2015) revealed 1,212 (84.1%) complete, 75 (5.2%) fragmented and 153 (10.7%) missing BUSCO gene models—a discrepancy caused by limitations of the models used by TransDecoder. We proceeded with inferring fragmentation on the dataset of all 57,252 TransDecoder putative protein-coding sequences. The dataset can be downloaded from https://doi.org/10.5281/zenodo.3628622.

5.3.2 Phylogenetic heuristics: few candidates, high proportion of ambiguous predictions

In order to investigate the performance of the phylogenetic heuristics on the transcriptome data, we applied them to the set of putative cassava protein-coding sequences identified by TransDecoder (Haas and Papanicolaou, 2016) including reference putative protein sequences from *Arabidopsis thaliana* and four other putative plant proteomes. We ran the 'bottom-up' variant of the GETHOGs algorithm (Train *et al.*, 2017) with default settings and obtained 21,412 input putative protein families for the heuristics. Relying on TransDecoder annotation, the GETHOGs algorithm retained only one putative isoform of each putative cassava gene. We applied the collapsing heuristic with a threshold of 0.95, and the likelihood ratio heuristic with significance levels of 0.2, 0.15, 0.1, 0.05 and 0.01. Predictions, both unambiguous and ambiguous, were validated against the reference putative

proteome of *Manihot esculenta* cv. AM560-2 (Bredeson *et al.*, 2016) using BLAST+ v2.2.30 (Camacho *et al.*, 2009).

On this particular input data, the collapsing approach did not manage to detect any fragmentation, while the likelihood ratio heuristic (LRH) detected a fair number of split putative transcripts ranging from 26 to 57 (Table 5.1) but only one prediction was unambiguous regardless of the heuristic significance level. Only one pair of fragments could be confirmed as a correct prediction. Consequently, the precision of the LRH (formula 3.2) on this dataset was unacceptably low: between 0.029 and 0.053 depending on the significance level of the heuristic.

61 cases	#splits	couldn't validate	could validate	correct	wrong	precision
Collapsing						
0.95	0	0	0	0	0	NA
LRH						
0.2	26	7	19	1	18	0.053
0.15	32	11	21	1	20	0.048
0.1	39	12	27	1	26	0.037
0.05	49	19	30	1	29	0.033
0.01	57	23	34	1	33	0.029

Table 5.1: Number of predictions obtained by the heuristics classified by the outcome of the BLAST+ (Camacho *et al.*, 2009) validation against reference putative cassava proteome (Bredeson *et al.*, 2016).
Collapsing branches with SH-like support less than 0.95 did not yield any predictions. In the consecutive experiment, we performed the likelihood ratio heuristic with significance levels of 0.2, 0.15, 0.1, 0.05 and 0.01. While the number of predictions was satisfactory given the number of cases subjected to the heuristics, the precision of the heuristic was very low.

One of the barriers to the higher number of reliable predictions was that out of 61 pairs subjected to heuristics, 59 belonged to the same putative hierarchical orthologous group (HOG (Train *et al.*, 2017)). Furthermore, the putative protein family contained 272 (45 cassava) putative sequences from cassava and five reference plants which indicates high fragmentation or gene duplication rate. The remaining two candidates were found in two distinct HOGs; one comprising 88 (31 cassava) and the other 548 (11 cassava) putative protein sequences. Putative cassava protein sequences in the latter two might not appear as fragmented as in the previously mentioned HOG, yet given the number of putative sequences from each species, it might be challenging to reject the possibility that fragments are not derived

from paralogous genes or from different isoforms of the same gene. Given its conservative performance on the wheat and wild olive data (sections 3.3, 4.3), it is little surprising that the collapsing approach did not yield any predictions. The redundancy in the input family also affected performance of the likelihood ratio heuristic which could not make more predictions (if there are more *bona fide* split transcripts).

The list of predictions for each approach and the choice of parameters can be downloaded from https://doi.org/10.6084/m9.figshare.11734293.v1.

5.3.3 ESPRIT and ESPRIT+LRH: more predictions, more ambiguous hits

In the study on bread wheat genome data (see Chapter 3) ESPRIT (Dessimoz *et al.*, 2011) yielded mainly complementary predictions to the ones obtained in the phylogenetic framework (section 3.3.4). Assuming similar behaviour on transcriptome data, we ran the software with default parameters on the same set of cassava and reference putative protein sequences as used for the collapsing heuristic and the likelihood ratio heuristic.

ESPRIT (Dessimoz *et al.*, 2011) unambiguously identified 456 pairs coming from the same transcript as well as 816,527 ambiguous predictions. The tool does not discard any putative isoform sequences and treats them as being coded by separate genes. Hence, some predictions included two putative isoform sequences (already) assigned to the same putative gene. Furthermore, as TransDecoder (Haas and Papanicolaou, 2016) sometimes predicts and annotates multiple coding sequences within a single Trinity isoform (Grabherr *et al.*, 2011), ESPRIT considered such cases for testing, too (our phylogenetics methods did not), and reported at least some of them[50] as fragments of the same putative transcript model. Regardless of their location (same isoform or not), we validated (BLAST+ v2.2.30

[50] We did not investigate ESPRIT's recall (Dessimoz *et al.*, 2011) on such cases as it was beyond the scope of the project.

(Camacho *et al.*, 2009)) all unambiguous predictions against the reference *Manihot esculenta* cv. AM560-2 putative proteome (Bredeson *et al.*, 2016). A breakdown of the predictions according to their Trinity annotation is shown in Table 5.2. Considering all 456 unambiguous predictions yielded high precision (formula 3.2)—0.842, while restricting only to the 235 pairs of sequences previously annotated as isoforms of different genes lowered precision to 0.715. A complete list of predictions IDs is available at https://doi.org/10.6084/m9.figshare.11734341.v1.

Finally, we assessed a meta-approach ESPRIT+LRH that takes a union of unambiguous predictions by ESPRIT (default parameters) (Dessimoz *et al.*, 2011) and the likelihood ratio heuristic (LRH) (significance level of 0.01). Again, we validated predictions the same way as for other approaches. By its definition, the approach has better ability to detect fragmentation than any of the two approaches alone and it identified 513 pairs of fragments, 292 of them representing isoforms of different putative genes. There were no predictions confirmed by both ESPRIT and LRH. Given the very low precision of the LRH approach, the precision of ESPRIT+LRH (calculated by formula 3.2) was lower than that of ESPRIT alone—0.752 and 0.584 depending whether all predicted pairs or just pairs coming from putative isoforms of different putative genes were considered.

ESPRIT								
Same putative isoform[51]			Different putative isoforms of the same putative gene			Different putative genes		
145			76			235		
Correct	Wrong	Cannot verify	Correct	Wrong	Cannot verify	Correct	Wrong	Cannot verify
106	0	39	21	2	53	103	41	91
ESPRIT + LRH (α=0.01)								
Same putative isoform[51]			Different putative isoforms of the same putative gene			Different putative genes		
145			76			292		
Correct	Wrong	Cannot verify	Correct	Wrong	Cannot verify	Correct	Wrong	Cannot verify
106	0	39	21	2	53	104	74	114

Table 5.2: The number of split transcript models unambiguously inferred by ESPRIT (Dessimoz *et al.*, 2011) and an approach combining ESPRIT with LRH.

Breakdown.

[51] Two distinct protein-coding sequences annotated by TransDecoder (Haas and Papanicolaou, 2016) within the same Trinity isoform model (Grabherr *et al.*, 2011)

5.4 Discussion and future work

In the following sections we briefly discuss the main outcomes of the tested approaches and provide an outlook on future work that has to be carried out in order to tailor our heuristics for use in improving fragmented transcriptome assemblies and their annotations.

5.4.1 Phylogenetic heuristics: marginal number of predictions

The pipeline developed for genome assemblies and annotations—inference of putative homologous families by GETHOGs (Train *et al.*, 2017), phylogenetic heuristics, methodology for evaluation of predictions—in its current state does not seem to be capable to deal with the challenges of the cassava transcriptome data. The dataset was not an easy one—cassava is a highly heterozygous diploid species which has likely undergone a whole genome duplication (section 1.3.4), and we used only five reference species for the inference which were estimated to have the shortest evolutionary distance to cassava among the available ones in the OMA Browser (Table D.1).

Recent comparative studies show that Trinity (Grabherr *et al.*, 2011) outperforms other *de novo* transcriptome assemblers, yet it still creates fragmented putative transcript sequences and outputs unrealistically high number of putative transcripts (Wang and Gribskov, 2017; Voshall and Moriyama, 2018; Hölzer and Marz, 2019). Also, some putative transcripts might be completely missing from the assembly (Voshall and Moriyama, 2018). Misassembled chimeric sequences tend to be created but to a lesser extent than in assemblies built from other evaluated *de novo* assemblers (Wang and Gribskov, 2017). The presence of fragmented putative transcript sequences, chimeric sequences, inversions, rearrangements, missing sequences and other artifacts in our cassava transcriptome assembly was reflected in 33.44% of paired-end input reads which could not be aligned to the assembly such that both ends align to the same contig. Although

assembly fragmentation is only one of the reasons for unaligned reads, we anticipated a larger number of predictions on corresponding fragmented putative protein-coding regions.

One of the key reasons for low number of predictions could be that in the process of computing putative input families for the heuristics, GETHOGs algorithm (Train *et al.*, 2017) chose only one putative protein isoform per putative gene and filtered out all others. This drastically reduced the overall theoretical number of all possible cases for heuristic investigation and the number of cases where the two isoform models were actually fragments of the same isoform model. Favourably, the chosen protein isoform model is the one for which GETHOGs finds indications that could have the highest number of putative evolutionary relationships within the dataset. Thus, such isoform models could potentially be more informative and provide more reliable information for the heuristics. So, the number of predictions might also indicate that the heuristics take a cautious approach toward making predictions to avoid suggesting unrealistic chimeric models.

Out of all 26-57 predictions made by the likelihood ratio heuristic, only one was unambiguous and validated as wrong. Among ambiguous predictions that could be validated, only one was classified as correct. This also potentially reveals some of the weaknesses of the validation process. First, a genome annotation might lack models of alternatively spliced variants, variants with alternative transcription start and alternative polyadenylation site (Grabherr *et al.*, 2011) which can prevent validation of some predictions. That could be the case here—in the putative cassava genome used for validation, each gene was represented by a single gene model, and a single putative protein sequence in the corresponding putative proteome. (On the other hand, if multiple reference isoforms are catalogued, models involved in predictions might have multiple equally good mappings which again does not allow for their validation in the current setting.) Second, some predictions involving fragments derived from heterozygous regions could be indeed wrong but they were maybe classified as wrong or not even validated if they were not sufficiently similar to the putative protein sequence predicted on the

region representation in the haploid-like reference genome assembly and annotation. Third, cassava is believed to have undergone a whole-genome duplication (WGD) (Bredeson *et al.*, 2016) and the heuristics do not seem to be capable to differentiate putative species-specific paralogous gene models from fragments of a single gene model (Piližota *et al.*, 2019)[52]. That probably led to some of the predictions validated as wrong. Yet, WGD maybe prevented validation of some pairs if the models under investigation mapped equally well to multiple protein models from duplicated genes. Finally, due to poly(A)+ selection protocol in transcriptome sequencing, it could be that some transcript models correspond to incompletely spliced pre-mRNA transcripts (see section 1.1.3). At their current state, the heuristics are not capable of differentiating between transcript models on the data derived from spurious pre-mRNA transcripts and transcript models on the data derived from mRNA transcripts. Indeed, a prediction combining the two is wrong. However, a transcript model stemming from pre-mRNA data might not have had a matching gene model in the putative cassava genome (Ingolia, 2014; Zhang *et al.*, 2015) and thus, the prediction could not be validated nor classified as wrong. Furthermore, a prediction involving a transcript model on a sequence from pre-mRNA and a transcript model on a sequence from mRNA could cause other correct predictions to be classified as ambiguous if they have a model in common. This is particularly problematic for correctly inferred fragmentation involving two transcript models derived from mRNA data which might be interpreted as less reliable just due to the ambiguity.

In terms of the absolute number of predictions and precision, ESPRIT (Dessimoz *et al.*, 2011) was more successful in this setting. It considered all cassava putative protein isoforms which provides a larger search space for

[52] In Piližota *et al.* (2019), random fragments were derived from 200 cassava gene models ("split gene models") and 200 pairs of putative paralogous cassava gene models ("paralogous gene models") from cassava assembly v4.1 downloaded from Phytozome v7 (Goodstein *et al.*, 2012). 75% of fragmented putative paralogs were species-specific, i.e. not shared by other 16 species in the dataset. The precision of the combined approach (collapsing@0.95 ∩ LRH@0.01) was 40%. When the pairs derived from species-specific putative paralogous gene models were excluded, the precision increased to 65%.

fragmented models. That led to only 0.06% of predictions classified as unambiguous. Also, as it relies on pairwise relationships, it is more resistant to partial or conflicting signals from multiple sequences and spurious signals from assembling artifacts which contributes to better performance in terms of precision. However, there is still a risk of making predictions involving fragments derived from different isoforms.

With respect to the validation of all methods, perhaps more pairs might have been validated and confirmed as correct if the reference assembly had been of the same cassava cultivar and less fragmented.

5.4.2 Directions for future work using this dataset

Given the outcome of collapsing approach and precision of the likelihood ratio heuristic on this dataset, the pipeline requires thorough investigation and probably major modifications to yield reliable predictions on the transcriptome data. The heuristics are indeed the key step in the inference, yet the confounding effect of pre-testing process should not be neglected as results depend on the evolutionary information contained in the corresponding putative protein families and on the putative sequences subjected to the heuristics (or not subjected to the heuristics as seen in section B.6 for the wheat data). Here we provide guidelines for future work that tackle all stages of the inference process—from predicting protein-coding regions on contigs to subjecting to phylogenetic heuristics. We hope they can shed light on how the input data responds to each of the building blocks in the pipeline and yield solutions for improvement.

One could start by mapping all Trinity (Grabherr *et al.*, 2011) contigs to the putative cassava reference genome (Bredeson *et al.*, 2016). Here one of the following scenarios can happen: i) whole contig maps to the putative genome and seems to contain only full length putative transcript sequences—a case which cannot be further improved by the heuristics; ii) a contig seems to contain a fragment of a putative transcript sequence—if other fragment(s)

can be found in the dataset, they should all be monitored along the pipeline; iii) a contig or its part(s) cannot be mapped to the putative reference genome—a case of uninformative (sub)sequences for this purpose.

The next step in the current setting involves running TransDecoder (Haas and Papanicolaou, 2016) which may fail to recognise some of the protein-coding sequences. This might mean omitting putative sequences which are indeed fragments. The performance of TransDecoder could be improved (e.g. by including BLAST homology search (Camacho *et al.*, 2009) as open reading frame retention criteria) or a more optimal tool could be used for the same purpose.

In order to consider a pair of putative sequences for heuristic examination, they first have to be assigned to the same top-level hierarchical orthologous group. The GETHOGs pipeline (Train *et al.*, 2017) was not specifically adapted for dealing with fragmented data and some of the potential pairs will be separated across different hierarchical groups. Bigger HOGs can be constructed by modifying parameters of the GETHOGs algorithm. The only risk is, which was observed in the wheat study (sections 3.2.9 and 3.3.3), that this might increase the number of ambiguous predictions. For some studies that might be even more favourable, especially if there is additional data or expertise to resolve ambiguities.

Another criteria that a pair of putative protein sequences has to satisfy prior to heuristic examination is the length of their overlap in the multiple sequence alignment of the corresponding HOG (Train *et al.*, 2017). If their overlap is even slightly longer than the threshold (here 10% of their lengths), they will be discarded. In this particular dataset, input putative families for the phylogenetic heuristics contained 6,492 within-family pairs of putative cassava sequences, yet only 61 pairs passed the overlap criteria and were subjected to heuristics. Intuitively, this seems rather small number for a *de novo* putative plant transcriptome. This indicates that choosing an optimal overlap parameter—the one that retains a fair number of pairs, yet does not

induce too many ambiguous predictions—might be quite important for sensitivity of the heuristics.

Working with putative plant transcriptomes—i.e. having multiple putative isoforms, including those from paralogous genes, expressed at various levels from all homologous copies of chromosomes in the dataset—can become very challenging for the heuristics to untangle pairwise relationships among fragments and reconstruct full-length sequences. Attempting to handle these phenomena while reconstructing isofom sequences, assemblers make mistakes which were here already reflected in the fraction of BUSCO gene models (Simão *et al.*, 2015) found in multiple copies—37.3% on the Trinity (Grabherr *et al.*, 2011) contigs and 33.1% in the output of TransDecoder (Haas and Papanicolaou, 2016)[53]. Thus, isoforms could be additionally inferred and confirmed with other tools before proceeding with the rest of the pipeline. In the phylogenetic heuristics, we considered only one putative protein isoform sequence per putative gene chosen by GETHOGs algorithm (Train *et al.*, 2017). This substantially reduced the search space for fragmented transcript models. But even in the setup where a single putative protein isoform sequence per putative gene was considered, we can observe room for improvement. As already pointed out above, 59 out of 61 candidate pairs in this study were found in a single putative protein family. We believe that some of these sequences are indeed fragments derived from the same or paralogous genes. Furthermore, if they are derived from the same gene, they could be fragments of the same or different isoforms. Thus, the genome annotation problem of “being fragments of the same gene model”, can be further decomposed here into i) being fragments of the same isoform model, and ii) being fragments of two isoform models of the same gene model. One way to approach this problem at this stage could be to correct some of the mistakes using evolutionary context by incorporating the possibility that fragments come from different isoforms of the same gene. More precisely, when in doubt whether putative sequences are from isoforms or paralogs, putative paralogy could be concluded if it is shared by other species in the

[53] Also observed for Trinity in a cross-species comparison of assemblers by Hölzer and Marz (2019)

reference dataset. We could also consider all assembled and annotated isoform sequences and examine them separately two by two. First, all putative isoform sequences could be assigned to putative protein families. Then, prior to testing, all putative isoform sequences expressed by the same putative genes as sequences under examination could be filtered from the putative protein family in order to reduce harming redundancy. Further improvement could be achieved by merging isoform models into a single gene model. An appropriate way of doing this is a topic in itself.

Finally, it is important to evaluate obtained predictions. In the current setting, validation is limited by the quality of the particular haploid-like cassava genome assembly (Bredeson *et al.*, 2016). Perhaps, additional data could be obtained such as gene models assigned to heterozygous regions, models of all reconstructed isoforms or an updated genome assembly and annotation. Upcoming research projects might even generate a reference assembly of the same cultivar (98/0581). Predictions could be validated using other computational and experimental methods but that might be out of scope of the project.

The suggestions outlined above might require a lot of work given the size of the dataset. Therefore, one could consider to start with a preliminary study on a small subset of cases. The output of BUSCO assessment (Simão *et al.*, 2015) provides a good source of instances, in particular a list of fragmented transcript models and a list of duplicated models. Being predicted as near-universal single copy putative orthologs by definition, fragmented BUSCOs are our primary targets for the phylogenetic heuristics. At the time of writing, it is not well understood what happens with them along the pipeline. Are complementary fragments of fragmented transcript models present in the dataset? Are they all translated with TransDecoder? Does the GETHOGs algorithm (Train *et al.*, 2017) retain these putative isoform sequences or perhaps shorter ones? If complementary fragments are there, are they clustered in the same putative protein family? If yes, how much do they overlap? Does there appear to be a full-length reference sequence? If yes, are the fragments approximately equally evolutionary distant from the

reference? Is the corresponding protein tree well resolved? The second group of informative BUSCOs—the duplicated ones—are likely misannotated isoforms of the single-copy genes and could be crucial for developing coping mechanisms for that phenomena. We are positive that a careful case-by-case analysis of both fragmented and duplicated transcript models indicated by BUSCO could already provide meaningful insights for pipeline improvements.

5.4.3 Directions for future work using a higher quality dataset

With this project, we aimed to make a contribution towards improving transcriptome annotation as well as towards facilitating crops, in particular cassava, research. As it turns out, it might be more fruitful to address one problem at a time starting with adjusting the heuristics to the transcriptome data.

One idea could be to take a well annotated high-quality transcriptome assembly and use it as a test case for developmental purposes. To avoid tackling two problems at the time—transcriptome data and low-quality data—the heuristics could be first employed to identify fragmentation in the high-quality target. Although this could provide insights into the behaviour of the heuristics on the transcriptome data, the experiment might as well yield too few predictions to make valid conclusions. However, the results might be useful in a larger analysis described as follows. A dataset of high-depth RNA sequencing reads coming from the same species (and ideally the same study that provided the high-quality assembly) could be subsampled at various lower depths. Then each subsample could be assembled, annotated, translated and subjected to the heuristics (e.g. as described in sections 5.2.2-5.2.3). Predictions could be validated against putative high-quality genome or proteome (like in section 5.2.4). The study could help understanding the behaviour of the heuristics depending on the sequencing depth and data fragmentation, and reveal necessary adjustments to the transcriptome data.

A good test case target species for the heuristics development could be *Arabidopsis thaliana*—the best annotated plant according to Bolger, Arsova and Usadel (2018). It is a small flowering plant with a small ~135 Mbp long diploid genome organised into five chromosomes. Its reference genome assembly (Lamesch *et al.*, 2011) omitting any redundant regions is 119,146,348 bp long (Phoenix Bioinformatics Corporation, 2010) of which 118,960,704 bp are ungapped (EMBL-EBI, 2020). The assembly contains only 100 contigs (N50 11,194,537 bp) organised into 5 scaffolds (N50 23,459,830 bp) (EMBL-EBI, 2020). There are 27,655 annotated protein-coding genes (Cheng *et al.*, 2017) with a reference dataset of 74,194 non-redundant putative transcripts (Zhang *et al.*, 2017). The datasets comprise comprehensively annotated data of high quality and thus, could facilitate the investigation and improvement of the heuristics.

Chapter 6: Conclusion

Genome and transcriptome sequencing and assembling have become integral parts of biological research—from a source of supporting evidence in basic research to key datasets in cutting-edge technological innovations. Despite their wide application and development in the past decades, assembling approaches still usually yield draft assemblies which are subsequently annotated with many fragmented gene models (Schliesky *et al.*, 2012; Denton *et al.*, 2014; Richards, 2018). In this PhD thesis, we aimed to establish a new framework for detecting fragments of the same gene or transcript model based on phylogenetic inference across closely related species. We assume that the minority of predictions is due to mistakes in annotation. For the majority of predictions, we anticipate that fragmentation already existed in the corresponding assembly (possible reasons described in section 1.1.3). Thus, the predictions can facilitate improving the assembly quality as well.

Homology, the key concept in comparative genomics, is fundamental for phylogeny reconstruction (section 1.4). The process of identifying putative homologs typically starts with comparing all-against-all putative protein sequences within and across species in a dataset of interest, as described in section 1.4.3. As this scales quadratically in terms of the number of sequences analysed, this step can become a bottleneck, thus limiting the number of putative proteomes that can be simultaneously analysed. Therefore, our first research contribution was towards developing a new, faster method for homology inference (Chapter 2). In this project, we explored ways of speeding up the all-against-all step while maintaining its sensitivity (Wittwer *et al.*, 2014). Aiming to implement transitivity of homology and, crucially, incorporating the concept of putative subsequence homology (concept described in section 1.4.2), our proof-of-concept resulted in a 4x speedup while recovering >99.6% of all putative homologs identified by the full all-against-all procedure on the sets of putative protein sequences.

Despite already providing a significant speedup with only a slight decrease in recall, the new method for homology inference could still be further improved. Unidentified putative homologous relationships translate into missed putative orthologous relationships, both bearing importance in downstream analyses, as indicated throughout section 1.4. Since putative homologs are missed due to putative sequences being placed in different clusters, future projects could investigate ways of merging clusters, choosing better cluster representatives and even consider representing clusters with, for example, profiles or hidden Markov model profiles. Further improvements in recall could be achieved by optimising alignment score and coverage thresholds, using a different substitution matrix, processing putative proteomes and protein sequences in a different order or clustering putative protein sequences based on sequence features. A lower number of clusters would decrease the time required to assign a new sequence to the existing clusters. Clusters could be merged based on members in common, or a modified clustering approach could be implemented which clusters only sequences within a species, and then merges clusters across species following a bottom-up traversal of the species tree. In cases where there are indications that two clusters could be evolutionary very distant, it might be worth to take a risk and conclude that inclusion in one cluster automatically implies omission from the other cluster, and avoid some computations. Finally, our work lays the foundation for further speedup that can be achieved by parallel implementation of the algorithm, which is currently work in progress by other lab members. Reducing the computational time, either with a serial or parallel implementation of the algorithm, is of particular importance for large datasets where the time to process all-against-all comparisons among all putative protein sequences within and across species can now be done in a fraction of the time, cutting down on other accompanying costs as well.

In the subsequent project, the centrepiece of the thesis, we tackled the problem of the fragmented annotation of plant genome assemblies (Chapter 3). We developed two phylogenetic heuristics—one that collapses branches having SH-like support below a chosen threshold, and another that exploits a likelihood ratio value. Assuming reliable full-length reference gene models

and corresponding putative protein sequences, the heuristics attempt to distinguish fragments derived from the same gene, as opposed to fragments derived from paralogous genes. We extensively validated these methods by 1) introducing and recovering fragmentation on the putative protein sequences of gene models assigned to bread wheat, *Triticum aestivum* cv. Chinese Spring, putative chromosome 3B (Choulet *et al.*, 2014); 2) by applying the methods to the putative protein sequences of gene models assigned to low-quality 3B assembly (International Wheat Genome Sequencing Consortium (IWGSC), 2014) and validating predictions against the putative proteome constructed on the high-quality 3B assembly (Choulet *et al.*, 2014); and 3) by comparing the performance of the proposed methods to the performance of existing methods, namely Ensembl Compara (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020) and ESPRIT (Dessimoz *et al.*, 2011). We suggest applying both heuristics to the data of interest and taking the intersection of their predictions as two lines of evidence that indicate fragmentation. Regarding the choice of phylogenetic reconstruction tool and its parameters, the heuristics demonstrated robustness to the tested settings of selected tools (FastTree v2.1.7 default installation (Price, Dehal and Arkin, 2010), FastTree v2.1.10 double-precision installation and RAxML v8.2.12 (Alexandros Stamatakis, 2014)).

Nevertheless, it is important to keep in mind the limitations of our approaches for detecting fragmentation, notably difficulties in dealing with fragments derived from close paralogs and difficulties with fragments derived from the same gene, yet from parts that have evolved at different evolutionary rates. In the first case, putative protein fragments often do not contain enough information to be distinguished as putative paralogs and the heuristics suggest merging corresponding gene models into a single one. Although biologically wrong, such gene models can still be informative in some applications, especially if two paralogs are identical at the protein sequence level. In the second case, due to the discrepancy in the evolutionary rates, the heuristics do not recognize fragments as parts of the same gene model. Here, an implementation of a more realistic evolutionary model would enable recovering some of the missed cases. Finally, the approaches depend on the

input putative protein families, multiple sequence alignment and tree building tools, and their parameters, all of which could be further explored and fine-tuned for a dataset under investigation.

Having evaluated and analysed the performance of the developed phylogenetic heuristics, we chose a set of parameters for application to the putative protein sequences of gene models constructed on a draft shotgun assembly of the entire bread wheat genome (International Wheat Genome Sequencing Consortium (IWGSC), 2014). Aiming for higher confidence of predictions, we considered only hits confirmed by both heuristics. The application revealed 1,221 unambiguous pairs of gene models which are, according to the heuristics, likely to be fragments of the same gene model. We hope their inclusion in the future studies will aid wheat and crop research, as well as wider comparative genomics community. We would also like to acknowledge that the availability of the reference bread wheat genome (International Wheat Genome Sequencing Consortium (IWGSC) *et al.*, 2018) now allows validation of our predictions and could facilitate further improvements of the heuristics.

In the subsequent project we examined the performance of the heuristics on a different dataset where only a single genome assembly and annotation were available for the target species (Chapter 4). More precisely, we attempted to shed light on fragmented annotation of the genome assembly of wild olive, *Olea europaea* var. *sylvestris* (Unver *et al.*, 2017), employing the developed heuristics with their default set of parameters. Considering only cases where both heuristics were in agreement, 102 unambiguous pairs of gene models could be merged into longer models.

Beside improving genome annotation of the wild olive assembly, the study was important for two methodological reasons. First, we provide readers with a step-by-step assessment of the data and methods. Namely, given a dataset of putative target proteome and reference proteomes, we showcase how to 1) investigate the behaviour of the heuristics on the dataset and decide whether to pursue detecting fragmentation with our approaches, 2)

decide which putative protein families could be too challenging for the inference methods, and 3) assess the plausibility of the predictions. We acknowledge limitations of the proposed examinations as they consider only limited information. Nonetheless, they can evidently help in dismissing some of the very challenging cases to prevent false positive predictions, as well as bring more evidence to support the credibility of inferred predictions. The process involves decision-making on the user side, yet considering all known and unknown biological and technical complexities involved, there is still a lot to be learnt before creating a unique automated framework for assessments, if possible at all. Second, the work supports applicability of the approaches beyond putative genome of bread wheat. Moreover, it proves its usefulness in circumstances where only a single genome assembly and annotation are available for the species of interest.

Although motivated by fragmentation in plant genomes, our phylogenetic heuristics are not plant-specific and can be applied to any species of interest. To ease and encourage their use, we provide the source code for the approaches allowing users to modify parameters as discussed in the study. The tool also outputs additional information related to the fragments and putative protein families involved in predictions which can facilitate examining the behaviour of the heuristics on a particular dataset and the assessment of predictions. If a GFF file for the target annotation is provided, it is updated with respect to the obtained predictions. The tool will hopefully become even more user-friendly in its future releases.

In the last research project, we tackled the problem of fragmented annotation of transcriptome assemblies (Chapter 5). Transcriptomes contain transcripts expressed at different levels, in different cells and at different developmental stages of an organism which challenges the assembly process and subsequently affects structural annotation, as seen in sections 1.1.3 and 1.2. However, having analogous problems on different types of data encouraged us to explore the behaviour of the heuristics developed for genomic data in the transcriptome landscape. We were primarily interested in the extent of their applicability to the annotated transcriptome assemblies and ways to

adapt them to deal with peculiarities of the transcriptome data. As a test case for the study, we assembled *de novo* a reasonably good transcriptome of *Manihot esculenta* cv. 98/0581 using Trinity (Grabherr *et al.*, 2011) and constructed putative peptide sequences with TransDecoder (Haas and Papanicolaou, 2016). We applied both phylogenetic heuristics and validated the obtained predictions against the reference set of *Manihot esculenta* cv. AM560-2 putative protein sequences (Bredeson *et al.*, 2016). Unfortunately, the heuristics did not rise to the challenge of transcriptome data; collapsing could not identify any split transcript models, while the likelihood ratio heuristic could (26-57 depending on the significance level) but with surprisingly low precision ranging from 0.029 to 0.053. Again as in the previous project, we compared these approaches to the pairwise approach ESPRIT (Dessimoz *et al.*, 2011) whose predictions were validated the same way as for the phylogenetic heuristics. ESPRIT made 456 unambiguous predictions with precision of 0.842 proving its pairwise-based inference superior to the phylogenetic heuristics on this particular dataset.

Unfortunately, due to the time constraints of the project, the outcomes of the transcriptome study are limited and not conclusive. We believe that the performance of the phylogenetic approaches cannot be explained only through their limitations observed on the genome data. Clearly, it seems that they require an in-depth investigation and major improvements before being reliably applicable to the transcriptome data. Perhaps the biggest challenge in the transcriptome setting is the appropriate treatment of putative isoforms as their redundancy poses challenges in inferring pairwise relationships among sequences. It can be particularly hard to correctly distinguish fragments derived from different isoforms of paralogous genes. Although the heuristics themselves are the backbone of the pipeline employed for identifying split transcript models, their sensitivity and precision depend on the input putative protein data. Therefore, we strongly encourage gaining a better understanding of all building blocks of the pipeline. For more details please see sections 5.4.2 and 5.4.3 where we attempted to provide well founded suggestions for future analyses that could tackle each step in the

pipeline and help reveal ways to improve the inference of fragmented transcript models in transcriptome assemblies.

With this thesis we contribute to the development of phylogenetic methods for detecting fragmented gene and transcript models in annotated genome and transcriptome assemblies respectively. We developed two phylogenetic heuristics which yielded 1,221 unambiguous predictions on the putative bread wheat genome and 102 unambiguous predictions on the putative wild olive genome. We provide the source code which can be used on the researcher's species of interest. We started exploring the capability of the heuristics on the annotation of cassava transcriptome which we assembled *de novo* with Trinity (Grabherr *et al.*, 2011) and annotated using TransDecoder (Haas and Papanicolaou, 2016). Despite their theoretical nature, we hope that our suggestions for future work provide a good starting point towards adjustments of the heuristics to the transcriptome data. As the approaches rely on putative homologous relationships, we also contributed to the project on speeding up homology inference by involving transitivity of putative subsequence-level homology (concept explained in section 1.4.2) which yielded 4x speedup and achieved >99% accuracy on the empirical datasets.

We hope that the outcomes of this research are a stepping stone towards the routine application of phylogenetic approaches in improving fragmented assemblies and their annotation.

Appendix A

Speeding up homology inference

A.1 Clustering strategy development

A.1.1 Examination of various clustering strategies

Table A.1: Randomly chosen putative bacteria proteomes used in preliminary analysis of various clustering strategies.

The data was downloaded from OMA database (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018), March 2014 release.

OMA 5-letter code	Taxon ID	Species name	Source	Release
CORAD	583355	Coraliomargarita akajimensis (strain DSM 45221 / IAM 15411 / JCM 23193 / KCTC 12865)	Genome Reviews	07-JUN-2011 (Rel. 130, Last updated, Version 7)
ECOLX	1040638	Escherichia coli	NCBI	AFOB02000109.1 GI:340738205
MYCMS	272632	Mycoplasma mycoides subsp. mycoides SC (strain PG1)	Genome Reviews	11-SEP-2007 (Rel. 80, Last updated, Version 74)

(table continues on the next page)

(table continues from the previous page)

OMA 5-letter code	Taxon ID	Species name	Source	Release
THEM4	391009	Thermosipho melanesiensis (strain BI429 / DSM 12029)	Genome Reviews	05-FEB-2008 (Rel. 86, Last updated, Version 2)
THET1	525904	Thermobaculum terrenum (strain ATCC BAA-798 / YNP1)	Genome Reviews	15-JUN-2010 (Rel. 122, Last updated, Version 6)

Table A.2: Preliminary analysis of various clustering strategies.
As a strategy was tested, new ideas were driven by its results. All strategies were tested on a dataset comprised of putative proteomes listed in Table A.1. Runtime and recall for all strategies is shown in Figure A.1.

Strategy	Description	Recall (%)
a)	Full OMA all-against-all (baseline) (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019)	100
b)	Cluster founding sequence is a cluster representative. Assign a putative sequence to only one cluster—first found such that *alignment-score(sequence, representative) > 181* (181 is a threshold in the existing OMA pipeline).	87.16

(table continues on the next page)

(table continues from the previous page)

Strategy	Description	Recall (%)
c)	Sort putative proteomes according to the number of putative protein sequences. Process the largest putative proteome first. Cluster founding sequence is a cluster representative. Assign a sequence to all clusters where *alignment-score(sequence, representative) > 181*.	96.94
d)	Cluster founding sequence is the initial cluster representative. Assign a putative sequence to all clusters where *alignment-score(sequence, representative) > 181*; always keep the longest putative sequence in the cluster as a representative. We did not investigate reasons for low recall compared to other strategies but a possible explanation could lie in changing cluster representative in this manner as it could cause cluster content to diverge from the putative sequence that introduced the cluster. Hence, similar putative sequences might not end up in the same cluster.	89.19
e)	Sort putative proteomes according to the number of putative protein sequences. Process the smallest putative proteome first. Cluster founding sequence is a cluster representative. Assign a putative sequence to all clusters where *alignment-score(sequence, representative) > 181*.	94.91

(table continues on the next page)

(table continues from the previous page)

Strategy	Description	Recall (%)
f)	Sort putative proteomes according to the number of putative protein sequences. Process the largest putative proteome first. First three putative sequences in the cluster are cluster representatives. Assign a putative sequence to all clusters where *alignment-score(sequence, representative) > 181* for at least one representative.	99.53
g)	Same as f) but with keeping track of alignment scores for already computed pairs. Hence, when a pair (sequence, cluster representative) is found in multiple clusters, its alignment score is computed only once.	99.53
h)	Sort putative proteomes according to the number of putative protein sequences. Process the largest putative proteome first. First three putative sequences in the cluster are cluster representatives. Assign a putative sequence to all clusters where *alignment-score(sequence, representative) > 135.75* for at least one representative. Smaller alignment score threshold yields bigger clusters as well as lower number of clusters since more sequences get assigned to already existing clusters.	99.25

(table continues on the next page)

(table continues from the previous page)

Strategy	Description	Recall (%)
i)	Sort putative proteomes according to the number of putative protein sequences. Process the largest putative proteome first. Cluster founding putative sequence is a cluster representative. Assign a putative sequence to all clusters where *alignment-score(sequence, representative) > 135.75*. If an entire length of a sequence (minus 20 amino acid residues tolerance) is not covered by any representative of the assigned clusters, found a new cluster with the sequence under examination as its cluster representative.	99.70
j)	Sort putative proteomes according to the number of putative protein sequences. Process the largest putative proteome first. First three putative sequences in the cluster are cluster representatives. Assign a putative sequence to all clusters where *alignment-score(sequence, representative) > 135.75* for at least one representative. If an entire length of a sequence (minus 20 amino acid residues tolerance) is not covered by any representative of the assigned clusters, found a new cluster with the sequence under examination as its cluster representative.	99.97

Since cluster content depends on its representative(s), and cluster representatives depend on the species and sequence order within the putative proteomes, we tried clustering on a dataset where putative proteomes are ordered according to the number of putative sequences while putative sequences within putative proteomes keep the same ordering as in the original database file. We tried two simple ideas using *ordered* proteomes—starting with a putative proteome with the highest number of putative protein sequences (Table A.2 c)) and starting with a putative proteome with the smallest number of putative protein sequences (Table A.2

e)). In both cases we assigned one representative per cluster and we assigned putative sequences to all clusters where *alignment-score(sequence, representative) > 181*—the same alignment score threshold as in the full all-against-all OMA approach (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019). The variant where we started with processing the largest putative proteome first achieved higher recall (96.94% vs. 94.91%). Being aware that this is just an indication of algorithm behaviour which also depends on the particular dataset, we proceeded with implementing sorting and starting with processing the largest putative proteome first into our final pipeline.

The majority of our preliminary analyses used the OMA (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019) alignment score threshold of 181 but we also explored what happens in the case of lower threshold. The reasoning behind those experiments was that lower threshold could yield bigger clusters (as more putative sequences get included) and possibly smaller number of clusters (as fewer putative sequences would not be assigned to any clusters and hence, would found a new cluster). Thus, perhaps fewer putative homologs would be missed, i.e. the recall would increase. We compared cluster variants with alignment score threshold 181 (Table A.2 f)) and 135.75 (=¾ * 181; arbitrarily chosen) (Table A.2 h)). In both cases we ordered putative proteomes according to the number of putative sequences and processed them starting with the largest, used 3 cluster representatives and assigned sequences to all clusters where the score criterion was satisfied. At first surprisingly, the variant with lower threshold yielded lower recall on this dataset (99.25% vs. 99.53%). Further analysis of the pairs it missed (section A.1.2 below) revielded that with incorporating sequence coverage as an additional criterion for creating new clusters, the recall increased to 99.97% (Table A.2 j)) or 99.70% if only one cluster representative was used (Table A.2 i)).

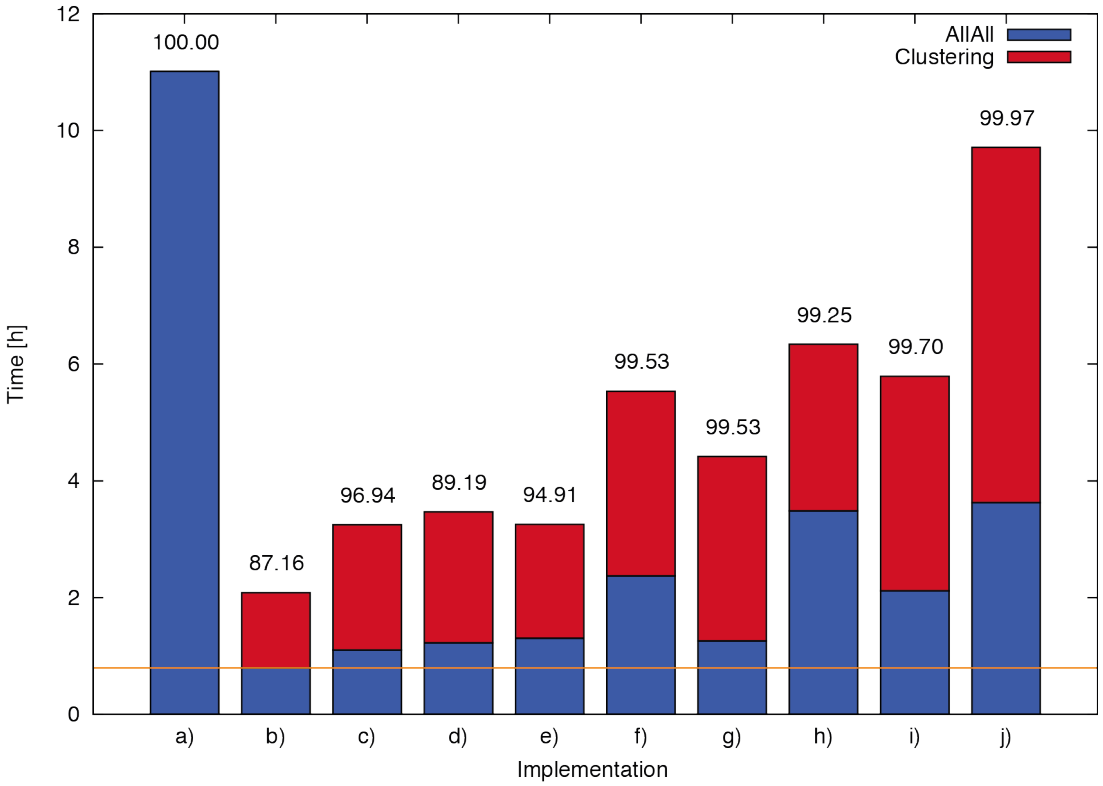


Figure A.1: Runtime and recall of various clustering strategies. Runtime of the all-against-all phase is depicted in blue while runtime of the clustering phase is depicted in red. The orange line marks (minimum) OMA (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019) runtime to calculate only pairwise alignments of all OMA putative homologs. a) OMA (baseline), b) 1 cluster representative, 1 cluster per putative sequence, c) process putative proteomes in descending order of the number of their putative proteins, 1 cluster representative, multiple clusters per putative protein sequence, d) cluster representative is the longest sequence in the cluster, multiple clusters per putative sequence, e) process putative proteomes in ascending order of the number of their putative protein sequences, 1 cluster representative, multiple clusters per putative sequence, f) same as c) but with 3 cluster representatives, g) same as f) but with keeping track of already computed pairwise alignments, h) same as f) but with lower alignment score threshold, i) process putative proteomes in descending order of the number of their putative protein sequences, 1 cluster representative, multiple clusters per sequence, lower alignment score threshold, coverage, j) same as i) but with 3 cluster representatives. More details on each strategy can be found in Table A.2.

A.1.2 Analysis of putative homologs missed by clustering strategy h) (Table A.2)

Implementation h) (order putative proteomes by the number of putative protein sequences, process putative proteomes in descending order of the number of their putative protein sequences, 3 cluster representatives, multiple clusters per sequence allowed, alignment score threshold 135.75) achieved recall of 99.25% and missed 247 putative homologs inferred by full OMA all-against-all approach (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2019). We had a look at 10 missed pairs (Table A.3).

Table A.3: Ten putative homologs missed by a clustering strategy on sorted putative proteomes (processing the largest putative proteome first) using 3 cluster representatives and assigning putative sequences to all clusters where they satisfy lower alignment score threshold (135.75).

Sequence 1 ID	Sequence 2 ID	Alignment score	Comments
MYCMS17	ECOLX2886	344.540	ECOLX2886 in cluster 739; MYCMS17 does not pass alignment score criterion to join any cluster
MYCMS277	ECOLX3049	184.044	ECOLX3049 in cluster 230; MYCMS277 does not pass alignment score criterion to join any cluster
MYCMS78	ECOLX2984	245.633	ECOLX2984 in cluster 1275; MYCMS78 does not pass alignment score criterion to join any cluster
MYCMS171	MYCMS618	2351.883	MYCMS171 in cluster 143; MYCMS618 does not pass alignment score criterion to join any cluster

(table continues on the next page)

(table continues from the previous page)

Sequence 1 ID	Sequence 2 ID	Alignment score	Comments
MYCMS751	MYCMS753	872.977	MYCMS751 in clusters 6326 and 6329; MYCMS753 does not pass alignment score criterion to join any cluster
MYCMS945	MYCMS947	248.777	MYCMS945 in clusters 1027, 3851 and 4922; MYCMS947 does not pass alignment score criterion to join any cluster
MYCMS105	MYCMS897	3005.401	MYCMS105 in cluster 2056; MYCMS897 does not pass alignment score criterion to join any cluster
MYCMS300	MYCMS854	330.991	MYCMS300 in clusters 84, 330, 1196, 1457, 1708 and 5831; MYCMS854 does not pass alignment score criterion to join any cluster
MYCMS760	ECOLX2984	192.502	MYCMS760 in clusters 6309 and 6556; ECOLX2984 in cluster 1275
MYCMS103	MYCMS912	874.091	MYCMS103 in cluster 587; MYCMS912 in cluster 6321

Eight out of ten pairs (first 8 in Table A.3) were missed due to low alignment score with cluster representatives at the time. In each case, one putative sequence was already assigned to clusters (not a cluster representative) and the other ended up founding a new cluster.

In one case (MYCMS760, ECOLX2984), reversing the ordering of processing putative proteomes could possibly lead to putting putative sequences to the same cluster (here we processed the putative proteome of *Escherichia coli* before the putative proteome of *Mycoplasma mycoides* subsp. mycoides SC (strain PG1)).

The most important insight came from the pair of putative sequences (MYCMS103, MYCMS912) and their MSA with representatives of cluster 587 (Fig. A.2). Sequence MYCMS103 and cluster representative CORAD1158 have a pairwise alignment score of 138.925. Thus, MYCMS103 becomes a member of cluster 587. However, the majority of its residues are different than those in the representatives. Sequence MYCMS912 spans the beginning of the alignment where it aligns well with MYCMS103 but not with cluster representatives. For example,

alignment-score(MYCMS912, MYCMS103) = 874.091,

alignment-score(MYCMS912, CORAD1158) = 34.438.

Sequence MYCMS912 does not become a member of the cluster and the pair (MYCMS103, MYCMS912) is missed. With introducing the coverage criterion (entire length of a putative sequence (minus 20 amino acid residues tolerance) covered by at least one representative), MYCMS103 would become a member of cluster 587 but would also found a new cluster. Sequence MYCMS912 would become a member of the new cluster as its alignment score with MYCMS103 is larger than 135.75.

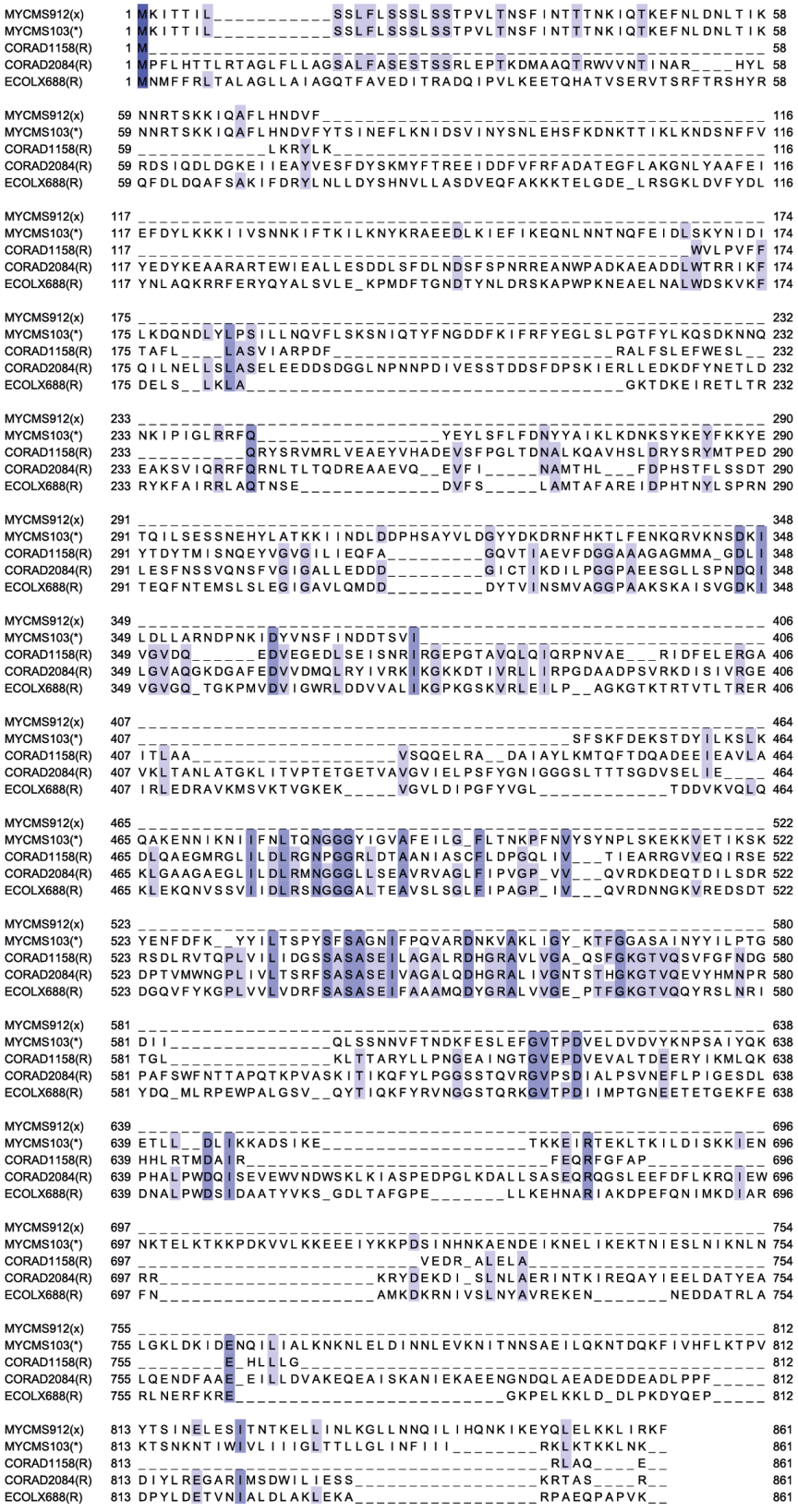


Figure A.2: MSA of MYCMS103, MYCMS912 and representatives of cluster 587.

Alignment drawn with JalView (Waterhouse *et al.*, 2009).

A.2 Datasets

Table A.4: Bacteria dataset.

OMA 5-letter code	Taxon ID	Species name	Source	Release
ACIC5	240015	Acidobacterium capsulatum (strain ATCC 51196 / DSM 11244 / JCM 7670)	Genome Reviews	16-JUN-2009 (Rel. 107, Last updated, Version 1)
CATAD	479433	Catenulispora acidiphila (strain DSM 44928 / NRRL B-24433 / NBRC 102108 / JCM 14897)	Genome Reviews	09-FEB-2010 (Rel. 117, Last updated, Version 5)
CHIPD	485918	Chitinophaga pinensis (strain ATCC 43595 / DSM 2588 / NCIB 11800 / UQM 2034)	Genome Reviews	09-FEB-2010 (Rel. 117, Last updated, Version 5)
CORAD	583355	Coraliomargarita akajimensis (strain DSM 45221 / IAM 15411 / JCM 23193 / KCTC 12865)	Genome Reviews	07-JUN-2011 (Rel. 130, Last updated, Version 7)

(table continues on the next page)

(table continues from the previous page)

OMA 5-letter code	Taxon ID	Species name	Source	Release
ECOLX	1040638	Escherichia coli	NCBI	AFOB02000109.1 GI:340738205
MYCMS	272632	Mycoplasma mycoides subsp. mycoides SC (strain PG1)	Genome Reviews	11-SEP-2007 (Rel. 80, Last updated, Version 74)
NAKMY	479431	Nakamurella multipartita (strain ATCC 700099 / DSM 44233 / JCM 9543 / Y-104)	Genome Reviews	10-AUG-2010 (Rel. 124, Last updated, Version 11)
ROTMD	680646	Rothia mucilaginosa (strain DY-18)	Genome Reviews	10-AUG-2010 (Rel. 124, Last updated, Version 5)
RHOSR	101510	Rhodococcus sp. (strain RHA1)	Genome Reviews	11-SEP-2007 (Rel. 80, Last updated, Version 20)
SALNS	423368	Salmonella newport (strain SL254)	Genome Reviews	25-NOV-2008 (Rel. 99, Last updated, Version 2)

(table continues on the next page)

(table continues from the previous page)

OMA 5-letter code	Taxon ID	Species name	Source	Release
SALTO	369723	Salinispora tropica (strain ATCC BAA-916 / DSM 44818 / CNB-440)	Genome Reviews	18-MAR-2008 (Rel. 88, Last updated, Version 1)
STRRD	479432	Streptosporangium roseum (strain ATCC 12428 / DSM 43021 / JCM 3005 / NI 9100)	Genome Reviews	25-MAY-2010 (Rel. 121, Last updated, Version 4)
THEM4	391009	Thermosipho melanesiensis (strain BI429 / DSM 12029)	Genome Reviews	05-FEB-2008 (Rel. 86, Last updated, Version 2)
THET1	525904	Thermobaculum terrenum (strain ATCC BAA-798 / YNP1)	Genome Reviews	15-JUN-2010 (Rel. 122, Last updated, Version 6)

Table A.5: Fungi dataset.

OMA 5-letter code	Taxon ID	Species name	Source	Release
ASPCL	344612	Aspergillus clavatus (strain ATCC 1007 / CBS 513.65 / DSM 816 / NCTC 3887 / NRRL 1)	EnsemblGenomes	Ensembl Fungi 16; CADRE; 19-OCT-2012
ASPFU	330879	Neosartorya fumigata (strain ATCC MYA-4609 / Af293 / CBS 101355 / FGSC A1100)	EBI	27-JUL-2005 (Rel. 84, Last updated, Version 2)
CANAW	294748	Candida albicans (strain WO-1)	EBI	12-JUN-2009 (Rel. 101, Last updated, Version 5)
EMENI	227321	Emericella nidulans (strain FGSC A4 / ATCC 38163 / CBS 112.46 / NRRL 194 / M139)	EnsemblGenomes	Ensembl Fungi v4; Eurofung Sep 2006; 17-FEB-2010
EURHE	41413	Eurotium herbariorum	JGI	JGI; Eurhe1; 12-MAR-2012

(table continues on the next page)

(table continues from the previous page)

OMA 5-letter code	Taxon ID	Species name	Source	Release
NEUCR	367110	Neurospora crassa (strain ATCC 24698 / 74-OR23-1A / CBS 708.71 / DSM 1257 / FGSC 987)	EnsemblGenomes	Ensembl Fungi 4; EF 1; 17-FEB-2010
PENCH	5076	Penicillium chrysogenum	JGI	JGI; Pench1; 07-MAR-2012
PENCW	500485	Penicillium chrysogenum (strain ATCC 28089 / DSM 1075 / Wisconsin 54-1255)	NCBI	NS_000201.1 GI:256353024
SCHPO	284812	Schizosaccharomyces pombe (strain 972 / ATCC 24843)	EnsemblGenomes	Ensembl Fungi 4; GeneDB EF 1; 17-FEB-2010
SPAPN	619300	Spathaspora passalidarum (strain NRRL Y-27907 / 11-Y1)	JGI	JGI; Spapa3; 07-MAR-2012
WALSE	148960	Wallemia sebi	JGI	JGI; Walse1; 07-MAR-2012

(table continues on the next page)

(table continues from the previous page)

OMA 5-letter code	Taxon ID	Species name	Source	Release
YEAST	559292	Saccharomyces cerevisiae (strain ATCC 204508 / S288c)	Ensembl	Ensembl 73; EF4; 23-AUG-2013

Table A.6: Mixed dataset.

OMA 5-letter code	Taxon ID	Species name	Source	Release
ARATH	3702	Arabidopsis thaliana	EnsemblGenomes	Ensembl Plants 20; TAIR10; 2-SEP-2013
BACSU	224308	Bacillus subtilis (strain 168)	Genome Reviews	12-SEP-2005 (Rel. 35, Last updated, Version 40)
HUMAN	9606	Homo sapiens	Ensembl	Ensembl 73; GRCh37; 24-AUG-2013
PLAF7	36329	Plasmodium falciparum (isolate 3D7)	EnsemblGenomes	Ensembl Protists release 2; PlasmoDB_5.5; 22-JUL-2009
XENTR	8364	Xenopus tropicalis	Ensembl	Ensembl 73; JGI_4.2; 23-AUG-2013

(table continues on the next page)

(table continues from the previous page)

OMA 5-letter code	Taxon ID	Species name	Source	Release
YEAST	559292	Saccharomyces cerevisiae (strain ATCC 204508 / S288c)	Ensembl	Ensembl 73; EF4; 23-AUG-2013

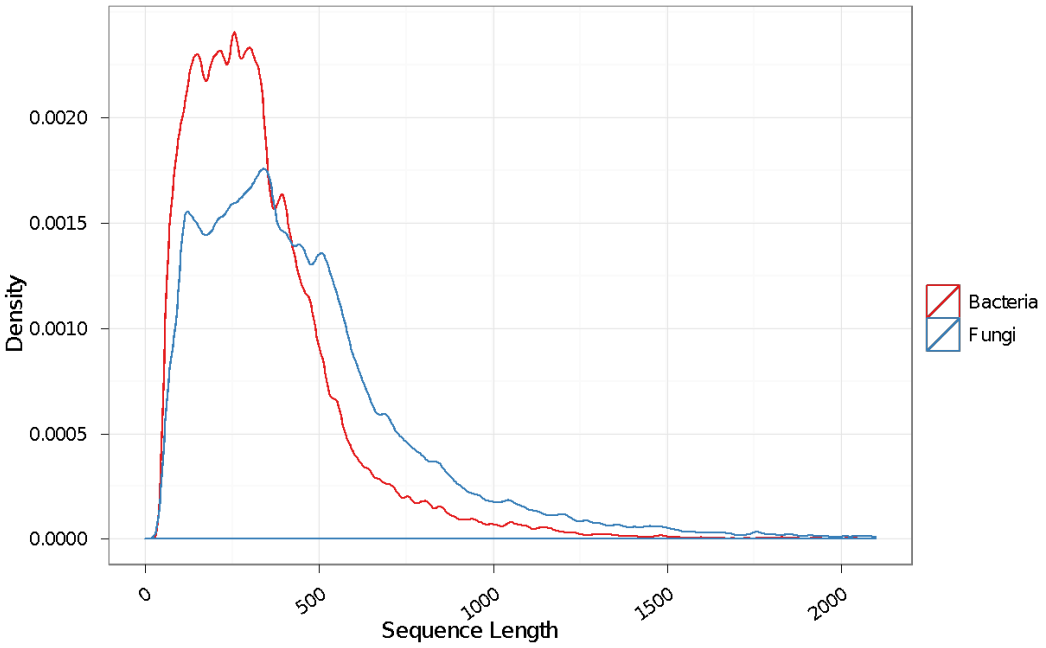


Figure A.3: Distribution of the sequence length (in number of amino acids) in bacteria and fungi datasets.

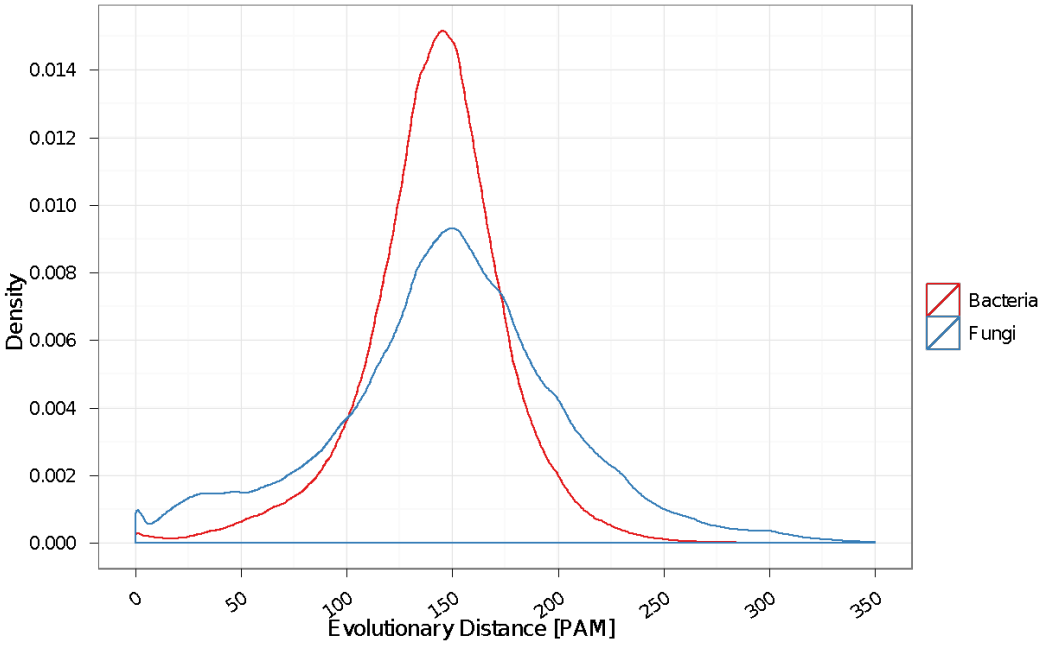


Figure A.4: Distribution of the estimated evolutionary distances (in PAM units) among putative homologous pairs inferred by full OMA all-against-all procedure (Roth, Gonnet and Dessimoz, 2008; Altenhoff *et al.*, 2014) in bacteria and fungi datasets.

A.3 Case studies of two missing putative homologous pairs

A.3.1 Example #1 (from bacteria dataset)

Putative sequences CHIPD1706 and CHIPD2153 (OMA IDs) have an alignment score of 2238.183 (estimated PAM distance: 40.3). CHIPD1706 is a member of a cluster with CHIPD533 (OMA ID) as a representative because the score 141.248 is above the threshold (137.75). CHIPD2153 however is not part of the cluster because the alignment score with the representative is 117.317 only and thus below the threshold. Figure A.5 depicts a multiple sequence alignment of these three putative sequences, and additionally three other cluster members. Figure A.6 depicts a phylogenetic tree of the sequences and confirms that the terminal branch of CHIPD2153 (outside) is slightly longer than that of CHIPD1706 (inside).

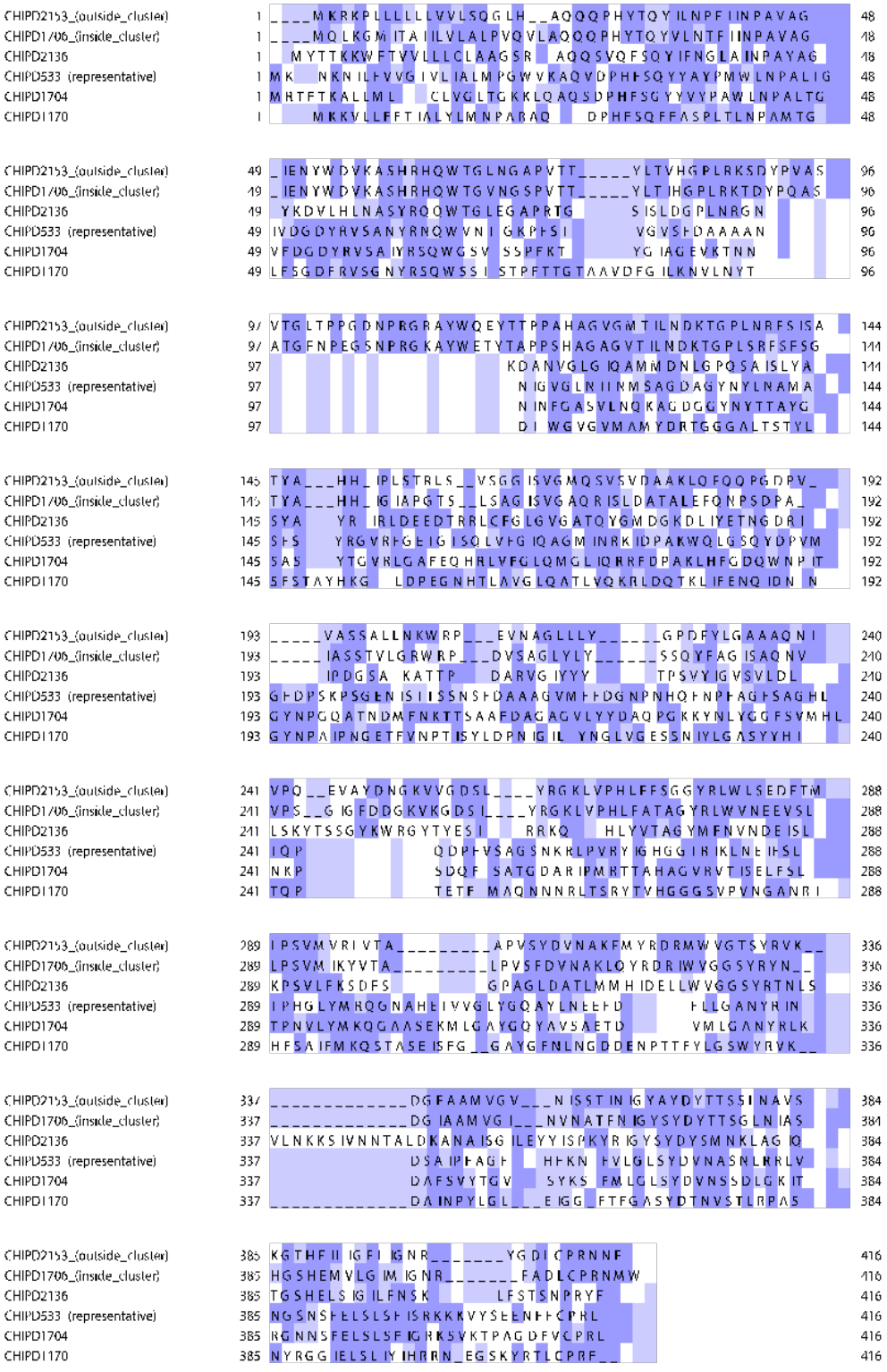


Figure A.5: Example #1: Multiple sequence alignment of the cluster to which CHIPD2153 should be included to recover the missing putative homologous pair CHIPD1706-CHIPD2153.

Alignment drawn with JalView (Waterhouse *et al.*, 2009)

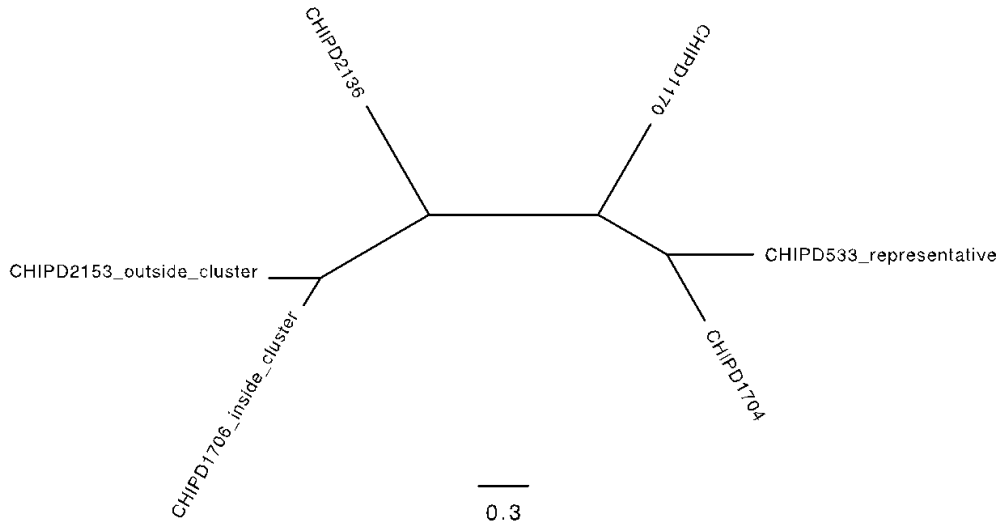


Figure A.6: Example #1: Estimated distance tree of the cluster and the missing putative sequence (CHIPD2153).

The corresponding MSA is provided in Figure A.5.

A.3.2 Example #2 (from fungi dataset)

Putative homology between putative sequences PENCW2854 (OMA ID) and PENCH3349 (OMA ID) is missed despite them being nearly identical (alignment score of 5,606.4 and estimated distance of 0.38 PAM units). PENCH2854 is a member of two clusters—the first cluster with representative PENCH2329 (OMA ID; score 138.9) and the second cluster with representative PENCW2605 (OMA ID; score 146.6). The alignment scores of PENCH3349 with the two representatives are below the threshold (124.6 and 128.9 respectively). Figures A.7 and A.8 provide representative subsets of the multiple sequence alignments for the two clusters.

We note that both clusters are very large: each contains >1000 putative sequences but only a small fraction of all member pairs are significant (7.8% and 26.96%). As mentioned in section 2.4.6, establishing mutually exclusive clusters might yield multiple smaller clusters instead and reduce the number of all-against-all comparisons.

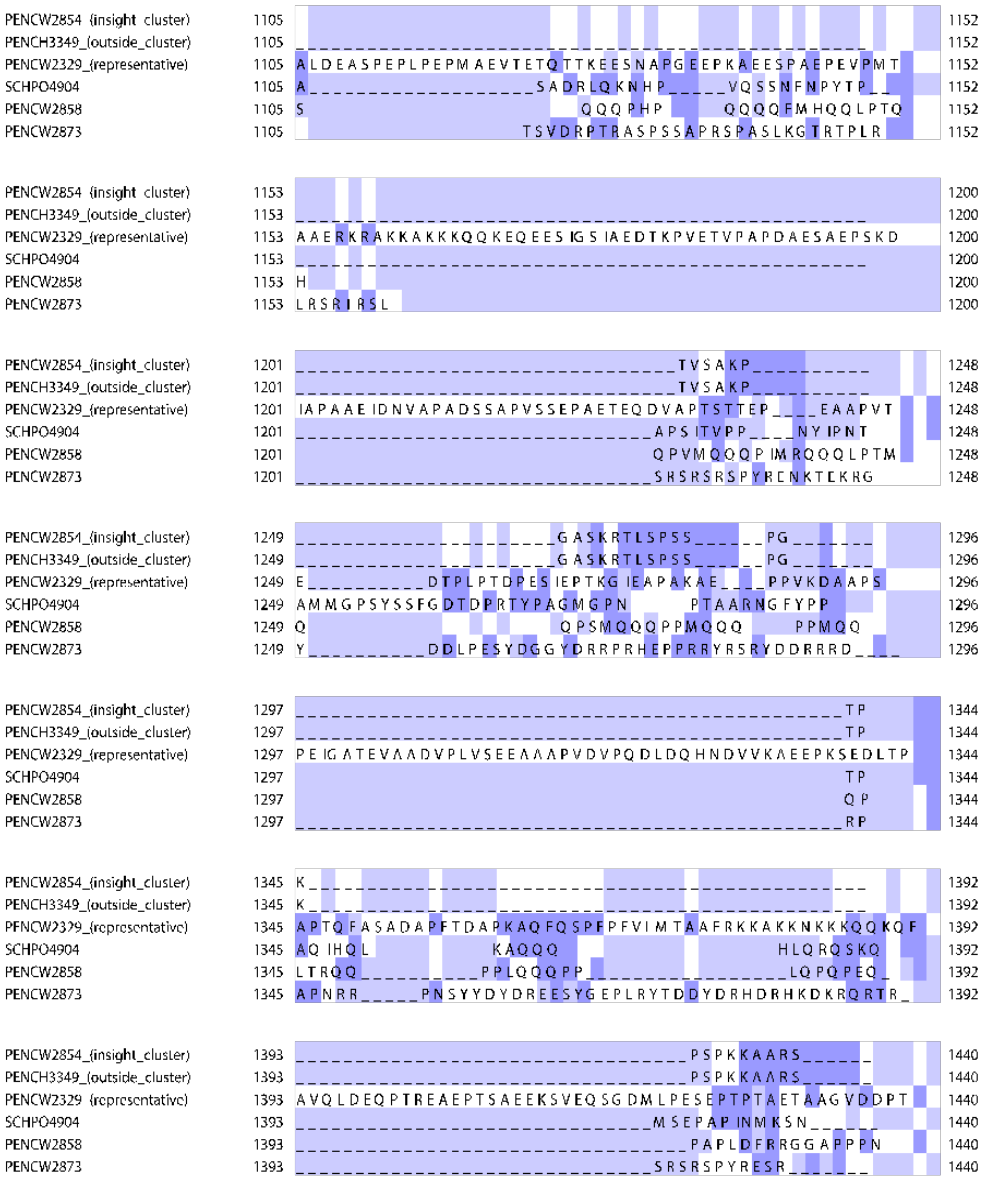


Figure A.7: Example #2: Representative extract of the multiple sequence alignment of the first cluster to which PENCH3349 should be included to recover the missing putative homologous pair PENCW2854-PENCH3349.

Alignment drawn with JalView (Waterhouse *et al.*, 2009).

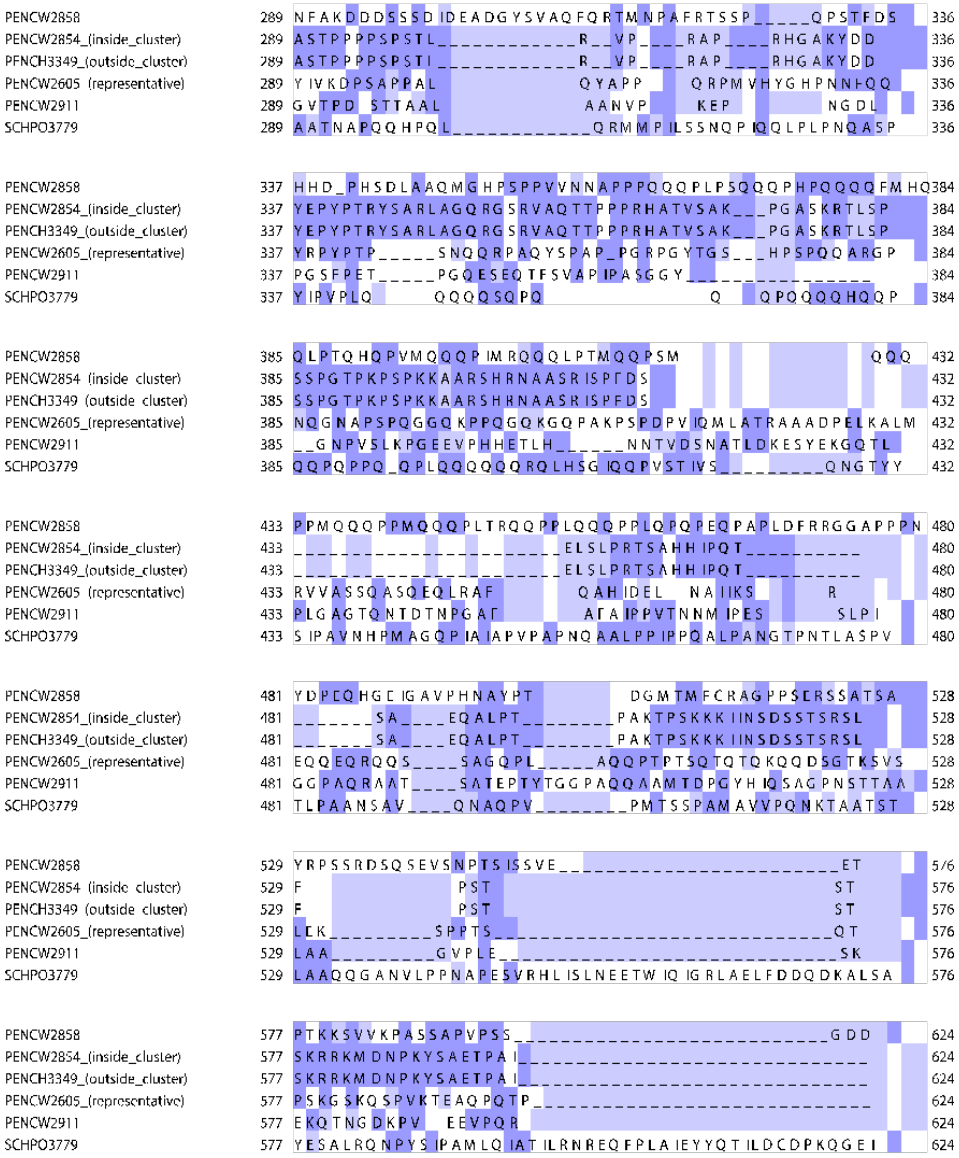


Figure A.8: Example #2: Representative extract of the multiple sequence alignment of the second cluster to which PENCH3349 should be included to recover the missing putative homologous pair PENCW2854-PENCH3349.

Alignments drawn with JalView (Waterhouse *et al.*, 2009).

A.4 Performance of the *k*-mer approaches

Table A.7: Runtimes in seconds for kClust (Hauser, Mayer and Söding, 2013) and UCLUST (Edgar, 2010) on the bacteria dataset (Table A.4) and fungi dataset (Table A.5).

	kClust `-s 1.12, -c 0.8`	**kClust** `-s 1.12, -c 0.5`	**UCLUST** `-id 0.3`	**UCLUST** `-id 0.3, -target_cov 0.5, -maxaccepts 0, -maxrejects 0`
Bacteria				
Run 1	26.20	26.53	3	12
Run 2	73.49	83.52	6	25
Run 3	124.79	133.87	11	68
Run 4	177.76	174.58	17	116
Run 5	343.36	310.16	39	186
Run 6	527.43	424.10	47	223
Fungi				
Run 1	146.1	317.18	9	54
Run 2	298.61	497.83	16	96
Run 3	512.43	804.16	22	157
Run 4	709.97	1138.86	28	190
Run 5	870.15	1274.76	34	276
Run 6	1075.51	1611.49	40	314

Table A.8: kClust (Hauser, Mayer and Söding, 2013): Recall for default settings (`-s 1.12 -c 0.8`) on the bacteria dataset (Table A.4) and fungi dataset (Table A.5).

	All-against-all	All-against-all only	All-against-all and kClust	kClust only		kClust
				Match length < 50 AA	Match length ≥ 50 AA	
Bacteria						
Run 1	12742	11906 (93.44%)	836 (6.56%)	2	442	1280
Run 2	61391	54739 (89.16%)	6652 (10.84%)	29	1068	7749
Run 3	139766	129181 (92.43%)	10585 (7.57%)	40	2111	12736
Run 4	318112	300606 (94.50%)	17506 (5.50%)	55	3714	21275
Run 5	909863	867961 (95.39%)	41902 (4.61%)	92	6920	48914
Run 6	1254733	1192481 (95.04%)	62252 (4.96%)	132	7571	69955
Fungi						
Run 1	58716	55460 (94.45%)	3256 (5.55%)	0	658	3914
Run 2	222788	205557 (92.27%)	17231 (7.73%)	3	2025	19259
Run 3	497128	456215 (91.77%)	40913 (8.23%)	4	2982	43899

(table continues on the next page)

(table continues from the previous page)

	All-against-all	All-against-all only	All-against-all and kClust	kClust only		kClust
				Match length < 50 AA	Match length ≥ 50 AA	
Run 4	920323	845476 (91.87%)	74847 (8.13%)	6	3642	78495
Run 5	1479326	1362942 (92.13%)	116384 (7.87%)	22	4630	121036
Run 6	2192589	2016956 (91.99%)	175633 (8.01%)	31	6189	181853

Table A.9: UCLUST (Edgar, 2010): Recall with parameters `-id 0.3, -target_cov 0.5, -maxaccepts 0, -maxrejects 0` on the bacteria dataset (Table A.4) and fungi dataset (Table A.5).

	All-against-all	All-against-all only	All-against-all and UCLUST	UCLUST only	UCLUST
Bacteria					
Run 1	12742	12252 (96.15%)	490 (3.85%)	270	760
Run 2	61391	56214 (91.57%)	5177 (8.43%)	807	5984
Run 3	139766	132412 (94.74%)	7354 (5.26%)	1353	8707

(table continues on the next page)

(table continues from the previous page)

	All-against-all	**All-against-all only**	**All-against-all and UCLUST**	**UCLUST only**	**UCLUST**
Run 5	909863	880522 (96.78%)	29341 (3.22%)	5227	34568
Run 6	1254733	1208408 (96.31%)	46325 (3.69%)	6066	52391
Fungi					
Run 1	58716	54844 (93.41%)	3872 (6.59%)	530	4402
Run 2	222788	201089 (90.26%)	21699 (9.74%)	2082	23781
Run 3	497128	447431 (90.00%)	49697 (10.00%)	3186	52883
Run 4	920323	828514 (90.02%)	91809 (9.98%)	4199	96008
Run 5	1479326	1337459 (90.41%)	141867 (9.59%)	5427	147294
Run 6	2192589	1977766 (90.20%)	214823 (9.80%)	7663	222486

Appendix B

Phylogenetic heuristics to identify fragments of the same gene model in low-quality putative genomes, with application to the putative wheat genome

B.1 Non-negativity of the likelihood ratio value T

Statement: In the following setting

H_s: *n-1* gene models (split gene model)

H_p: *n* gene models (gene models on sequences coming from paralogous genes)

$T = 2ln\frac{L(Hp)}{L(H_s)}$, where *L()* denotes the maximum estimator under each hypothesis

The likelihood ratio value is non-negative ($T \geq 0$).

Proof of statement: We can demonstrate this by contradiction.

Assume that $T < 0$, i.e. $2ln\frac{L(Hp)}{L(H_s)} < 0$. This implies $ln\frac{L(Hp)}{L(H_s)} < 0 \Rightarrow \frac{L(Hp)}{L(H_s)} < 1 \Rightarrow L(H_p) < L(H_s)$, i.e. that the maximum likelihood estimate (MLE) under H_s is greater than the MLE under H_p. But in this case, we can take the maximum likelihood (ML) tree under H_s, and add two terminal branches of length 0 at the leaf corresponding to the split gene model (depicted in Fig. 3.4 on the right). Under H_p, we can then assign each of the putative paralogous fragments to each of the new tips, and keep the remaining sequences identical. Now, consider that gaps are treated as missing data (the standard treatment in ML tree inference software), and thus the likelihood is obtained by integrating over every possible state for each gap character. As a result, with that tree, the likelihood under H_p should be identical to that under H_s, i.e.

$L(H_p) = L(H_s)$. So $T = 2ln\frac{L(Hp)}{L(H_s)} = 2ln1 = 2 * 0 = 0$, which contradicts the premise.

(Normally, the extra degrees of freedom afforded by decoupling the two fragments result in a higher likelihood of the ML tree under H_p, yielding a positive T.)

B.2 Datasets for simulations and validation

Table B.1: Putative proteomes exported from OMA Browser (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018) and used as input data for GETHOGs algorithm (Altenhoff *et al.*, 2013) in simulations.

The second column contains information on the database release that OMA Browser retrieved an assembly and annotation from.

Species	**Database**
Aegilops tauschii	Ensembl Plants 21
Arabidopsis thaliana	Ensembl Plants 20
Brachypodium distachyon	Ensembl Plants 21
Hordeum vulgare var. distichum	Ensembl Plants 16
Oryza brachyantha	Ensembl Plants 21
Oryza glaberrima	Ensembl Plants 21
Oryza sativa subsp. indica	Ensembl Plants 21
Oryza sativa subsp. japonica	Ensembl Plants v7
Setaria italica	Ensembl Plants 21
Sorghum bicolor	Sbi1_4
Triticum aestivum cv. Chinese Spring	Ensembl Plants 26
Triticum urartu	Ensembl Plants 19
Zea mays	Ensembl Plants v8

Table B.2: Putative proteomes exported from OMA Browser (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018) and used as input data for GETHOGs algorithm (Altenhoff *et al.*, 2013) in validation on *Triticum aestivum* cv. Chinese Spring chromosome 3B.

The second column contains information on the database release that OMA Browser retrieved an assembly and annotation from.

Species	**Database**
Aegilops tauschii	Ensembl Plants 21
Arabidopsis thaliana	Ensembl Plants 20
Brachypodium distachyon	Ensembl Plants 21
Hordeum vulgare var. distichum	Ensembl Plants 16
Oryza brachyantha	Ensembl Plants 21
Oryza glaberrima	Ensembl Plants 21
Oryza sativa subsp. indica	Ensembl Plants 21
Oryza sativa subsp. japonica	Ensembl Plants 27
Setaria italica	Ensembl Plants 21
Sorghum bicolor	Sbi1_4
Triticum aestivum cv. Chinese Spring	Ensembl Plants 21
Triticum urartu	Ensembl Plants 19
Zea mays	Ensembl Plants 27

B.3 Validation on 3B survey assembly

B.3.1 Less stringent validation

For each unambiguous or ambiguous prediction we:

1. Take the initial non-modified putative sequences of both fragments and BLAST+ (Camacho *et al.*, 2009) them against the high-quality putative sequences assigned to chromosome 3B (`-evalue 0.001`) (Choulet et al. 2014).
2. For each query sequence, identify BLAST+ hit(s) with the highest bitscore (`bitscore`). Keep only hit(s) with `qcovs` ≥ 95 and `pident` ≥ 95, if any. If there are no such hits for any of the queries, the pair cannot be validated.
3. For each query and its hits from 2., keep only hits with the highest `qcovs`. If there are multiple hits per query that satisfy the criteria, then filter out all hits with `pident` lower than the highest present. If there are still multiple hits for any of the queries, we consider that an ambiguous mapping and do not validate the pair.
4. If both queries have the same best hit, the prediction is considered to be correct. Otherwise, we consider it wrong.

B.3.2 More stringent validation

All steps are the same as in *B.3.1 Less stringent validation* except the step 2. Here, in addition to `qcovs` ≥ 95 and `pident` ≥ 95, we require all mismatches between a query and a hit to be at the ends of a query sequence.

Let's say that our tolerance length is M. Suppose that first N_1 and last N_2 positions of a query are not covered by a hit. If $N_1 > M$ or $N_2 > M$, then the hit does not pass the criteria. For a given query and a hit such that $0 \leq N_1$,

$N_2 \leq M$, consider their BLAST+ alignment (Camacho *et al.*, 2009). We allow mismatches to be only in the query's first M-N_1 or last M-N_2 aligned positions, and we set $M=5$.

B.4 Results of simulations

Table B.3: Results of simulations on putative protein families inferred by Ensembl pipeline (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020).

	#true positive	#false positive
Collapsing		
0.65	19	2
0.75	28	3
0.85	45	7
0.9	52	7
0.95	57	8
LRH		
0.2	80	45
0.15	85	51
0.1	89	62
0.05	94	71
0.01	98	78
Combination		
Coll 0.95 + LRH 0.2	49	6
Coll 0.95 + LRH 0.15	51	6
Coll 0.95 + LRH 0.1	54	7
Coll 0.95 + LRH 0.05	56	8
Coll 0.95 + LRH 0.01	57	8

Table B.4: Results of simulations on the top-level HOGs (Altenhoff *et al.*, 2013).

	#true positive	#false positive
Collapsing		
0.65	30	7
0.75	36	10
0.85	53	18
0.9	67	24
0.95	78	29
LRH		
0.2	51	17
0.15	59	20
0.1	72	27
0.05	79	32
0.01	89	39
Combination		
Coll 0.95 + LRH 0.2	43	14
Coll 0.95 + LRH 0.15	50	15
Coll 0.95 + LRH 0.1	59	20
Coll 0.95 + LRH 0.05	63	23
Coll 0.95 + LRH 0.01	72	26

B.5 Results of validation

Table B.5: Validation on annotated low-quality assembly of bread wheat chromosome 3B (GETHOGs (Altenhoff *et al.*, 2013) default settings, less stringent BLAST+ (Camacho *et al.*, 2009) validation).

GETHOGs default settings					
BLAST+ `pident` 95; `qcovs` 95					
	#splits	**couldn't validate**	**could validate**	**correct**	**wrong**
Collapsing					
0.65	14	8	6	6	0
0.75	19	10	9	9	0
0.85	37	21	16	16	0
0.9	54	31	23	23	0
0.95	73	39	34	31	3
LRH					
0.2	59	29	30	25	5
0.15	69	33	36	30	6
0.1	81	39	42	36	6
0.05	94	46	48	41	7
0.01	106	54	52	43	9
0.008	107	55	52	43	9
0.006	107	55	52	43	9
0.004	107	55	52	43	9
0.002	111	59	52	43	9
Combined					
Coll 0.95 + LRH 0.2	47	25	22	20	2
Coll 0.95 + LRH 0.15	52	26	26	23	3
Coll 0.95 + LRH 0.1	57	27	30	27	3
Coll 0.95 + LRH 0.05	62	29	33	30	3

(table continues on the next page)

(table continues from the previous page)

	#splits	couldn't validate	could validate	correct	wrong
Coll 0.95 + LRH 0.01	66	33	33	30	3
Coll 0.95 + LRH 0.008	66	33	33	30	3
Coll 0.95 + LRH 0.006	66	33	33	30	3
Coll 0.95 + LRH 0.004	66	33	33	30	3
Coll 0.95 + LRH 0.002	69	36	33	30	3

Table B.6: Validation on annotated low-quality assembly of bread wheat chromosome 3B (GETHOGs (Altenhoff *et al.*, 2013) default settings, more stringent BLAST+ (Camacho *et al.*, 2009) validation).

GETHOGs default settings					
BLAST+ `pident` 95; `qcovs` 95; ends 5					
	#splits	**couldn't validate**	**could validate**	**correct**	**wrong**
Collapsing					
0.65	14	10	4	4	0
0.75	19	13	6	6	0
0.85	37	25	12	12	0
0.9	54	37	17	17	0
0.95	73	49	24	24	0
LRH					
0.2	59	39	20	18	2
0.15	69	45	24	22	2
0.1	81	53	28	26	2

(table continues on the next page)

(table continues from the previous page)

	#splits	couldn't validate	could validate	correct	wrong
0.05	94	61	33	30	3
0.01	106	69	37	32	5
0.008	107	70	37	32	5
0.006	107	70	37	32	5
0.004	107	70	37	32	5
0.002	111	74	37	32	5
Combined					
Coll 0.95 + LRH 0.2	47	31	16	16	0
Coll 0.95 + LRH 0.15	52	34	18	18	0
Coll 0.95 + LRH 0.1	57	36	21	21	0
Coll 0.95 + LRH 0.05	62	38	24	24	0
Coll 0.95 + LRH 0.01	66	42	24	24	0
Coll 0.95 + LRH 0.008	66	42	24	24	0
Coll 0.95 + LRH 0.006	66	42	24	24	0
Coll 0.95 + LRH 0.004	66	42	24	24	0
Coll 0.95 + LRH 0.002	69	45	24	24	0

Table B.7: Validation on annotated low-quality assembly of bread wheat chromosome 3B (GETHOGs (Altenhoff *et al.*, 2013) relaxed settings, less stringent BLAST+ (Camacho *et al.*, 2009) validation).

GETHOGs default settings except `MinScore = 150; LengthTol = 0.4; ReachabilityCutoff = 0.3`					
BLAST+ `pident 95; qcovs 95`					
	#splits	**couldn't validate**	**could validate**	**correct**	**wrong**
Collapsing					
0.65	24	15	9	7	2
0.75	40	24	16	13	3
0.85	75	45	30	25	5
0.9	107	60	47	39	8
0.95	139	78	61	49	12
LRH					
0.2	168	96	72	45	27
0.15	187	107	80	51	29
0.1	213	120	93	55	38
0.05	250	143	107	64	43
0.01	287	165	122	72	50
0.008	290	167	123	72	51
0.006	292	167	125	72	53
0.004	296	169	127	73	54
0.002	301	172	129	75	54
Combined					
Coll 0.95 + LRH 0.2	86	45	41	31	10
Coll 0.95 + LRH 0.15	95	49	46	35	11
Coll 0.95 + LRH 0.1	105	55	50	38	12
Coll 0.95 + LRH 0.05	124	67	57	45	12
Coll 0.95 + LRH 0.01	133	74	59	47	12

(table continues on the next page)

(table continues from the previous page)

	#splits	couldn't validate	could validate	correct	wrong
Coll 0.95 + LRH 0.008	134	75	59	47	12
Coll 0.95 + LRH 0.006	134	75	59	47	12
Coll 0.95 + LRH 0.004	135	75	60	48	12
Coll 0.95 + LRH 0.002	136	75	61	49	12

Table B.8: Validation on annotated low-quality assembly of bread wheat chromosome 3B (GETHOGs (Altenhoff *et al.*, 2013) relaxed settings, more stringent BLAST+ (Camacho *et al.*, 2009) validation).

GETHOGs default settings except `MinScore = 150; LengthTol = 0.4; ReachabilityCutoff = 0.3`					
BLAST+ `pident 95; qcovs 95; ends 5`					
	#splits	**couldn't validate**	**could validate**	**correct**	**wrong**
Collapsing					
0.65	24	17	7	6	1
0.75	40	27	13	11	2
0.85	75	51	24	20	4
0.9	107	72	35	30	5
0.95	139	93	46	38	8
LRH					
0.2	168	121	47	33	14
0.15	187	134	53	38	15
0.1	213	152	61	41	20
0.05	250	182	68	46	22
0.01	287	213	74	51	23

(table continues on the next page)

(table continues from the previous page)

	#splits	couldn't validate	could validate	correct	wrong
0.008	290	215	75	51	24
0.006	292	215	77	51	26
0.004	296	217	79	52	27
0.002	301	221	80	53	27
Combined					
Coll 0.95 + LRH 0.2	86	55	31	25	6
Coll 0.95 + LRH 0.15	95	59	36	29	7
Coll 0.95 + LRH 0.1	105	66	39	31	8
Coll 0.95 + LRH 0.05	124	81	43	35	8
Coll 0.95 + LRH 0.01	133	88	45	37	8
Coll 0.95 + LRH 0.008	134	89	45	37	8
Coll 0.95 + LRH 0.006	134	89	45	37	8
Coll 0.95 + LRH 0.004	135	89	46	38	8
Coll 0.95 + LRH 0.002	136	90	46	38	8

B.6 Approximation to recall values

B.6.1 Procedure

1. BLAST+ v2.2.30 (Camacho *et al.*, 2009) all putative protein sequences of gene models on the low-quality genome assembly of chromosome 3B (International Wheat Genome Sequencing Consortium (IWGSC), 2014) against putative protein sequences of predicted genes on the high-quality genome assembly of chromosome 3B (`-evalue 0.001`) (Choulet *et al.*, 2014).
2. For every annotated protein sequence on the low-quality assembly obtain the best mapping using the chosen procedure—either less or more stringent as described above in sections B.3.1-B.3.2 (steps 2 and 3).
3. If two low-quality putative sequences have the same best mapping, we consider this as an indication of a split gene model. Hence, check if the pair was subjected to testing. Check outcomes of the tests, if any (Fig. B.1).
4. Calculate approximation to recall values:
   4.1. Considering only tested cases:

$$recall\ (tested) = \frac{\#Inferred\ splits}{\#Tested\ cases}$$

   4.2. Considering all pairs of sequences that have common best mapping:

$$recall\ (all) = \frac{\#Inferred\ splits}{\#Pairs\ of\ sequences\ with\ common\ mapping}$$

Combinations of BLAST+ (Camacho *et al.*, 2009) mapping and putative protein families used:

- BLAST+ with less stringent criteria, HOGs with default GETHOGs settings (Altenhoff *et al.*, 2013) on input data listed in Table B.2
- BLAST+ with more stringent criteria, HOGs with default GETHOGs settings on input data listed in Table B.2

- BLAST+ with less stringent criteria, HOGs with relaxed GETHOGs settings (`MinScore := 150, LengthTol := 0.4, ReachabilityCutoff := 0.3`) on input data listed in Table B.2
- BLAST+ with more stringent criteria, HOGs with relaxed GETHOGs settings (`MinScore := 150, LengthTol := 0.4, ReachabilityCutoff := 0.3`) on input data listed in Table B.2

We required that candidate putative sequences had a minimum length of 50 AA and mutually overlap less than 10% in the corresponding multiple sequence alignment. Each application of likelihood ratio heuristics used 500 bootstrap samples. As before, we ran Mafft v7.164b with default parameters (Katoh and Standley, 2013) to align families and FastTree v2.1.8 with default parameters (Price, Dehal and Arkin, 2010) to reconstruct phylogenetic trees.

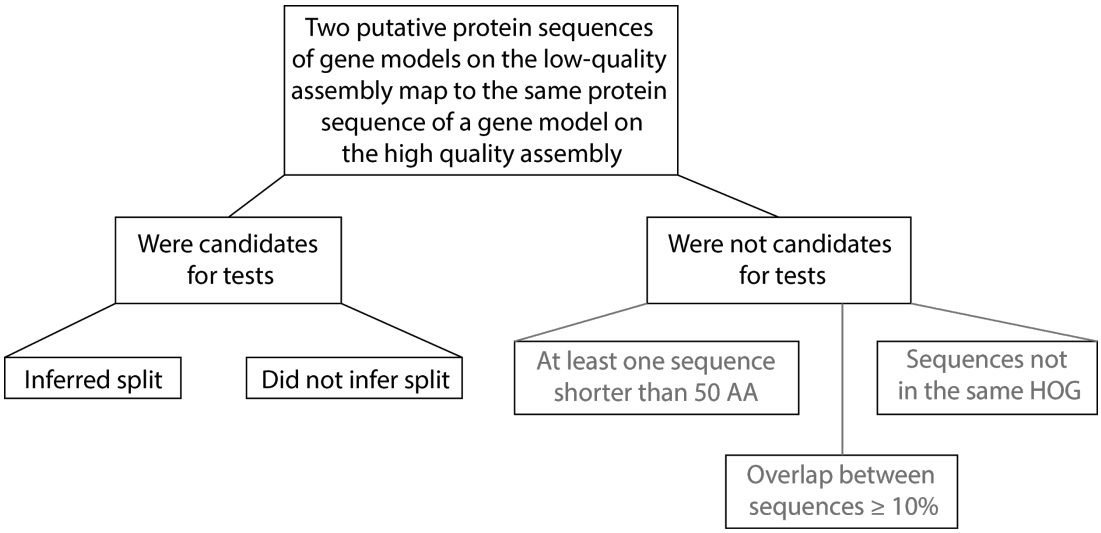


Figure B.1: Approximation to recall values.

For every pair of putative protein sequences corresponding to the low-quality assembly (International Wheat Genome Sequencing Consortium (IWGSC), 2014) that map to the same putative protein sequence corresponding to the high-quality assembly (Choulet *et al.*, 2014), we checked if they passed our pre-testing criteria (same putative protein family, length, overlap), and if yes, the outcomes of the tests. Approximations to recall values were then calculated by dividing the number of predictions by either the number of tested cases or the total number of pairs that map to the same high-quality putative sequence.

B.6.2 Results

Recall approximation on cases subjected to heuristics (Fig. 3.10a-c, Tables B.10-B.13, prediction IDs available at https://doi.org/10.6084/m9.figshare.11733609.v1) was consistent with recall on simulated fragmentation (Fig. 3.8, Table B.4). Yet, the fraction of subjected cases was low (~11.5-28.3%) (Table B.9) which led to low recall when all cases were considered (Fig. 3.10d-f, Tables B.10-B.13). Probably not all cases were indeed cases of fragmented gene models—some of them were perhaps paralogous, maybe even mapping to the same region of the corresponding high-quality reference model. However, given such a large

fraction of pairs not subjected to the heuristic inference, we decided to investigate this further.

A pair of putative protein sequences will not be considered for our heuristics if the two putative sequences are assigned to different putative protein families, if one or both sequences are shorter than 50 AA or if they overlap for ≥10% of their sequence length in the multiple sequence alignment of their putative protein family. In our experiments, only ~1.95-5.22% of cases not examined by heuristics were excluded due to short sequence(s) length (Table B.9). Further ~9.74-29.62% of unexamined cases were eliminated due to long mutual overlap in the alignment. Unexpectedly, ~66.98-88.31% of unexamined cases were left out from heuristic inference because the two putative sequences were not members of the same putative protein families. To gain better understanding why fragments were found in different HOGs (Altenhoff *et al.*, 2013), we performed a case-by-case analysis of randomly selected cases. Furthermore, to investigate the potential in applying heuristics to discarded cases, we ran our heuristics in the setting where putative protein families were obtained by running GETHOGs algorithm with default settings and merged HOGs for candidate putative sequences found in different ones. We checked predictions on the target cases obtained with both BLAST+ (Camacho *et al.*, 2009) less and more stringent mappings.

Table B.9: Counting the pairs that were and were not subjected to the heuristics.

Strikingly, the majority of cases were not tested due to putative sequences being placed in different putative protein families (HOGs) (Altenhoff *et al.*, 2013).

BLAST LESS STRINGENT, HOGs default				**BLAST MORE STRINGENT, HOGs default**			
Total number of draft pairs that map to the same reference putative protein				Total number of draft pairs that map to the same reference putative protein			
400				187			
Tested	Not tested			Tested	Not tested		
46 (11.5%)	354 (88.5%)			33 (17.65%)	154 (82.35%)		
	Different HOGs	Too short	Long overlap		Different HOGs	Too short	Long overlap
	285 (80.51 %)	7 (1.98%)	62 (17.51%)		136 (88.31 %)	3 (1.95%)	15 (9.74%)
BLAST LESS STRINGENT, HOGs relaxed				**BLAST MORE STRINGENT, HOGs relaxed**			
Total number of draft pairs that map to the same reference putative protein				Total number of draft pairs that map to the same reference putative protein			
400				187			
Tested	Not tested			Tested	Not tested		
76 (19%)	324 (81%)			53 (28.34%)	134 (71.66%)		
	Different HOGs	Too short	Long overlap		Different HOGs	Too short	Long overlap
	217 (66.98 %)	11 (3.4%)	96 (29.62%)		97 (72.39 %)	7 (5.22%)	30 (22.39 %)

Table B.10: Approximation to recall values on annotated low-quality assembly of bread wheat chromosome 3B (less stringent BLAST+ (Camacho *et al.*, 2009) mapping, GETHOGs (Altenhoff *et al.*, 2013) default settings).

BLAST LESS STRINGENT, HOGs default (BLAST+ `pident 95, qcovs 95`; GETHOGs default settings)				
	Split (TP)	**Paralogs (FN)**	**Recall on cases subjected to the heuristics**	**Recall on all**
Collapsing				
0.65	6	40	0.130	0.015
0.75	9	37	0.196	0.023
0.85	16	30	0.348	0.040
0.9	23	23	0.500	0.058
0.95	31	15	0.674	0.078
LRH				
0.2	25	21	0.543	0.063
0.15	30	16	0.652	0.075
0.1	36	10	0.783	0.090
0.05	41	5	0.891	0.103
0.01	43	3	0.935	0.108
0.008	43	3	0.935	0.108
0.006	43	3	0.935	0.108
0.004	43	3	0.935	0.108
0.002	43	3	0.935	0.108

(table continues on the next page)

(table continues from the previous page)

	Split (TP)	Paralogs (FN)	Recall on cases subjected to the heuristics	Recall on all
Combined				
Coll 0.95 + LRH 0.2	20	26	0.435	0.050
Coll 0.95 + LRH 0.15	23	23	0.500	0.058
Coll 0.95 + LRH 0.1	27	19	0.587	0.068
Coll 0.95 + LRH 0.05	30	16	0.652	0.075
Coll 0.95 + LRH 0.01	30	16	0.652	0.075
Coll 0.95 + LRH 0.008	30	16	0.652	0.075
Coll 0.95 + LRH 0.006	30	16	0.652	0.075
Coll 0.95 + LRH 0.004	30	16	0.652	0.075
Coll 0.95 + LRH 0.002	30	16	0.652	0.075

Table B.11: Approximation to recall values on annotated low-quality assembly of bread wheat chromosome 3B (more stringent BLAST+ (Camacho *et al.*, 2009) mapping, GETHOGs (Altenhoff *et al.*, 2013) default settings).

BLAST MORE STRINGENT, HOGs default (BLAST+ `pident 95, qcovs 95, ends 5`; GETHOGs default settings)				
	Split (TP)	**Paralogs (FN)**	**Recall on cases subjected to the heuristics**	**Recall on all**
Collapsing				
0.65	4	29	0.121	0.021
0.75	6	27	0.182	0.032
0.85	12	21	0.364	0.064
0.9	17	16	0.515	0.091
0.95	24	9	0.727	0.128
LRH				
0.2	18	15	0.545	0.096
0.15	22	11	0.667	0.118
0.1	26	7	0.788	0.139
0.05	30	3	0.909	0.160
0.01	32	1	0.970	0.171
0.008	32	1	0.970	0.171
0.006	32	1	0.970	0.171
0.004	32	1	0.970	0.171
0.002	32	1	0.970	0.171

(table continues on the next page)

(table continues from the previous page)

	Split (TP)	Paralogs (FN)	Recall on cases subjected to the heuristics	Recall on all
Combined				
Coll 0.95 + LRH 0.2	16	17	0.485	0.086
Coll 0.95 + LRH 0.15	18	15	0.545	0.096
Coll 0.95 + LRH 0.1	21	12	0.636	0.112
Coll 0.95 + LRH 0.05	24	9	0.727	0.128
Coll 0.95 + LRH 0.01	24	9	0.727	0.128
Coll 0.95 + LRH 0.008	24	9	0.727	0.128
Coll 0.95 + LRH 0.006	24	9	0.727	0.128
Coll 0.95 + LRH 0.004	24	9	0.727	0.128
Coll 0.95 + LRH 0.002	24	9	0.727	0.128

Table B.12: Approximation to recall values on annotated low-quality assembly of bread wheat chromosome 3B (less stringent BLAST+ (Camacho *et al.*, 2009) mapping, GETHOGs (Altenhoff *et al.*, 2013) relaxed settings).

BLAST LESS STRINGENT, HOGs relaxed (BLAST+ `pident 95, qcovs 95`; GETHOGs default settings except `MinScore = 150; LengthTol = 0.4; ReachabilityCutoff = 0.3`)				
	Split (TP)	**Paralogs (FN)**	**Recall on cases subjected to the heuristics**	**Recall on all**
Collapsing				
0.65	7	69	0.092	0.018
0.75	13	63	0.171	0.033
0.85	25	51	0.329	0.063
0.9	39	37	0.513	0.098
0.95	49	27	0.645	0.123
LRH				
0.2	45	31	0.592	0.113
0.15	51	25	0.671	0.128
0.1	55	21	0.724	0.138
0.05	64	12	0.842	0.160
0.01	72	4	0.947	0.180
0.008	72	4	0.947	0.180
0.006	72	4	0.947	0.180
0.004	73	3	0.961	0.183
0.002	75	1	0.987	0.188

(table continues on the next page)

(table continues from the previous page)

	Split (TP)	Paralogs (FN)	Recall on cases subjected to the heuristics	Recall on all
Combined				
Coll 0.95 + LRH 0.2	31	45	0.408	0.078
Coll 0.95 + LRH 0.15	35	41	0.461	0.088
Coll 0.95 + LRH 0.1	38	38	0.500	0.095
Coll 0.95 + LRH 0.05	45	31	0.592	0.113
Coll 0.95 + LRH 0.01	47	29	0.618	0.118
Coll 0.95 + LRH 0.008	47	29	0.618	0.118
Coll 0.95 + LRH 0.006	47	29	0.618	0.118
Coll 0.95 + LRH 0.004	48	28	0.632	0.120
Coll 0.95 + LRH 0.002	49	27	0.645	0.123

Table B.13: Approximation to recall values on annotated low-quality assembly of bread wheat chromosome 3B (more stringent BLAST+ (Camacho *et al.*, 2009) mapping, GETHOGs (Altenhoff *et al.*, 2013) relaxed settings).

BLAST MORE STRINGENT, HOGs relaxed (BLAST+ `pident 95, qcovs 95, ends 5`; GETHOGs default settings except `MinScore = 150; LengthTol = 0.4; ReachabilityCutoff = 0.3`)				
	Split (TP)	**Paralogs (FN)**	**Recall on cases subjected to the heuristics**	**Recall on all**
Collapsing				
0.65	6	47	0.113	0.032
0.75	11	42	0.208	0.059
0.85	20	33	0.377	0.107
0.9	30	23	0.566	0.160
0.95	38	15	0.717	0.203
LRH				
0.2	33	20	0.623	0.176
0.15	38	15	0.717	0.203
0.1	41	12	0.774	0.219
0.05	46	7	0.868	0.246
0.01	51	2	0.962	0.273
0.008	51	2	0.962	0.273
0.006	51	2	0.962	0.273
0.004	52	1	0.981	0.278
0.002	53	0	1.000	0.283

(table continues on the next page)

(table continues from the previous page)

	Split (TP)	Paralogs (FN)	Recall on cases subjected to the heuristics	Recall on all
Combined				
Coll 0.95 + LRH 0.2	25	28	0.472	0.134
Coll 0.95 + LRH 0.15	29	24	0.547	0.155
Coll 0.95 + LRH 0.1	31	22	0.585	0.166
Coll 0.95 + LRH 0.05	35	18	0.660	0.187
Coll 0.95 + LRH 0.01	37	16	0.698	0.198
Coll 0.95 + LRH 0.008	37	16	0.698	0.198
Coll 0.95 + LRH 0.006	37	16	0.698	0.198
Coll 0.95 + LRH 0.004	38	15	0.717	0.203
Coll 0.95 + LRH 0.002	38	15	0.717	0.203

B.6.3 Case-by-case analysis of randomly selected pairs with putative sequences found in different putative protein families

We investigated 8 randomly chosen pairs that were not subjected to heuristics (Table B.14) because the putative protein sequences were placed in different putative protein families (GETHOGs algorithm (Altenhoff *et al.*, 2013) with default set of parameters on the input data listed in Table B.2):

- 1 pair with a putative protein sequence shorter than 50 AA which would have predicted fragmented gene model by our heuristics (if examined)
- 1 pair with a putative protein sequence shorter than 50 AA which would have not predicted fragmented gene model by our heuristics
- 1 pair with mutual overlap of the putative protein sequences in the MSA of merged corresponding HOGs ≥10% which would have predicted fragmented gene model by our heuristics
- 1 pair with mutual overlap of the putative protein sequences in the MSA of merged corresponding HOGs ≥10% which would have not predicted fragmented gene model by our heuristics
- 2 pairs that would be subjected to heuristics if corresponding HOGs were merged and would have predicted fragmented gene models
- 2 pairs that would be subjected to heuristics if corresponding HOGs were merged and would have not predicted fragmented gene models

Both less and more stringent BLAST+ (Camacho *et al.*, 2009) mappings found indications that gene models involved in these 8 cases could be fragments of a longer gene model.

Table B.14: Randomly selected (previously) not tested cases and outcomes of our heuristics.

Sequence 1 ID; length; HOG ID (#sequences in HOG)	**Sequence 2 ID; length; HOG ID (#sequences in HOG)**	**Not tested due to**	**Outcomes** **CoII 0.95; LRH 0.01 (p-value); combined**
Traes_3B_163FC6BE5 47 AA HOG27481 (2)	Traes_3B_D1A0C478F 48 AA HOG3516 (26)	Different HOGs; in addition putative sequences too short	yes yes (0.81) yes
Traes_3B_DEB6C5A5C 77 AA HOG16006 (11)	Traes_3B_53E723173 38 AA HOG22410 (6)	Different HOGs; in addition one putative sequence too short	no yes (0.21) no
Traes_3B_6BEA473F3 210 AA HOG23413 (3)	Traes_3B_9ED6CC72 5 342 AA HOG1381 (4)	Different HOGs; in addition long overlap (0.9976; 0.7186)	yes yes (0.31) yes
Traes_3B_01F0A66DF 346 AA HOG22410 (6)	Traes_3B_E3185192A 268 AA HOG22409 (4)	Different HOGs; in addition long overlap (1.0; 0.7967)	no no (0.0099) no
Traes_3B_EAE6A3943 106 AA HOG9677 (2)	Traes_3B_CEEF8E71 A 181 AA HOG9689 (25)	Different HOGs	yes yes (0.64) yes
Traes_3B_98DDEDC49 311 AA HOG18118 (5)	Traes_3B_E1776A6B6 787 AA HOG18117 (21)	Different HOGs	yes yes (0.96) yes
Traes_3B_1235C7C8A 64 AA HOG25507 (12)	Traes_3B_854157B07 90 AA HOG19583 (20)	Different HOGs	no yes (0.98) no

(table continues on the next page)

(table continues from the previous page)

Sequence 1 ID; length; HOG ID (#sequences in HOG)	**Sequence 2 ID; length; HOG ID (#sequences in HOG)**	**Not tested due to**	**Outcomes** **CoII 0.95; LRH 0.01 (p-value); combined**
Traes_3B_DEB6C5A5C 77 AA HOG16006 (11)	Traes_3B_01F0A66DF 346 AA HOG22410 (6)	Different HOGs	no yes (0.14) no

For each case we had a look at multiple sequence alignment of merged HOGs and performed heuristic inference (collapsing with threshold 0.95, likelihood ratio heuristic with significance 0.01 and combination of the two).

In all cases except (Traes_3B_01F0A66DF, Traes_3B_E3185192A) (Fig. B.2), at least one HOG contained domains/regions not present in the other HOG (examples in Fig. B.3-B.5). This is to be expected since by its definition GETHOGs algorithm (Altenhoff *et al.*, 2013) would not group together putative protein sequences with significantly different[54] putative domain composition, i.e. putative sequences where indications of homology could not be found along most of their length. They would have been grouped together if the algorithm considered putative subsequence homology or putative subsequence orthology. In the case of (Traes_3B_01F0A66DF, Traes_3B_E3185192A)—the only selected case where neither of our heuristics would infer fragmentation—the picture is different (Fig. B.2). It is dubious whether to merge HOGs or not, and if yes, how to interpret the resulting set of putative sequences.

[54] "significantly different" defined by Altenhoff *et al.* (2013) based on shared putative domains and their lengths

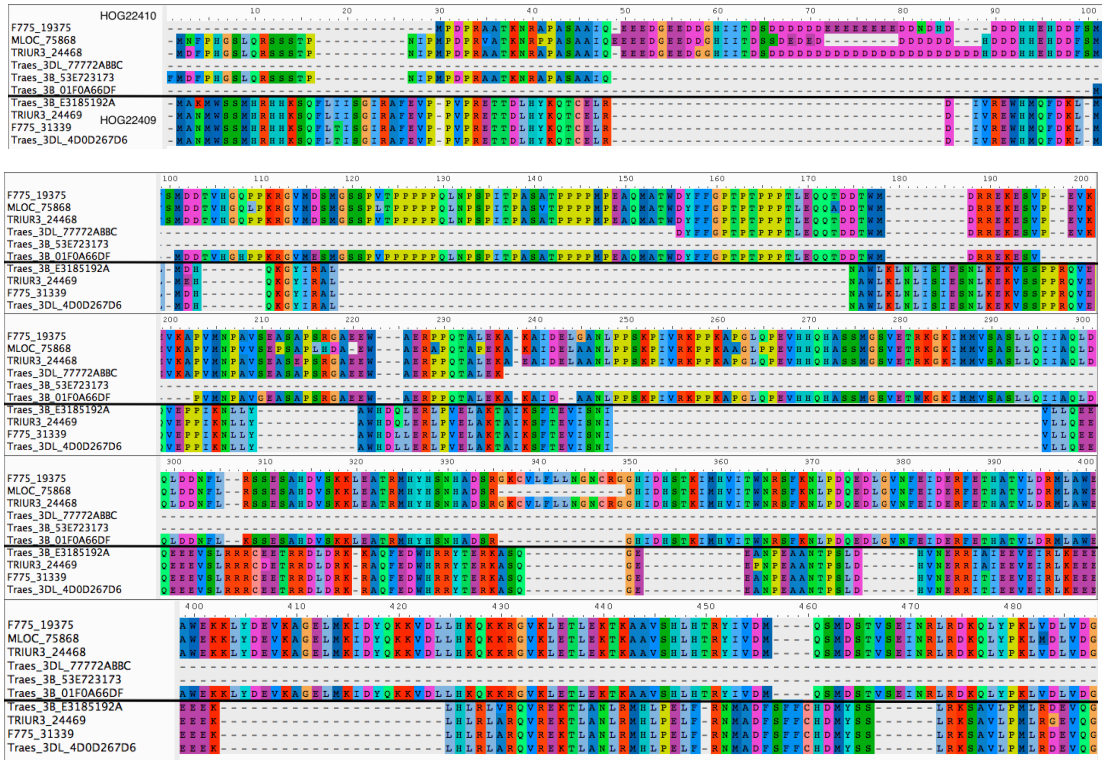
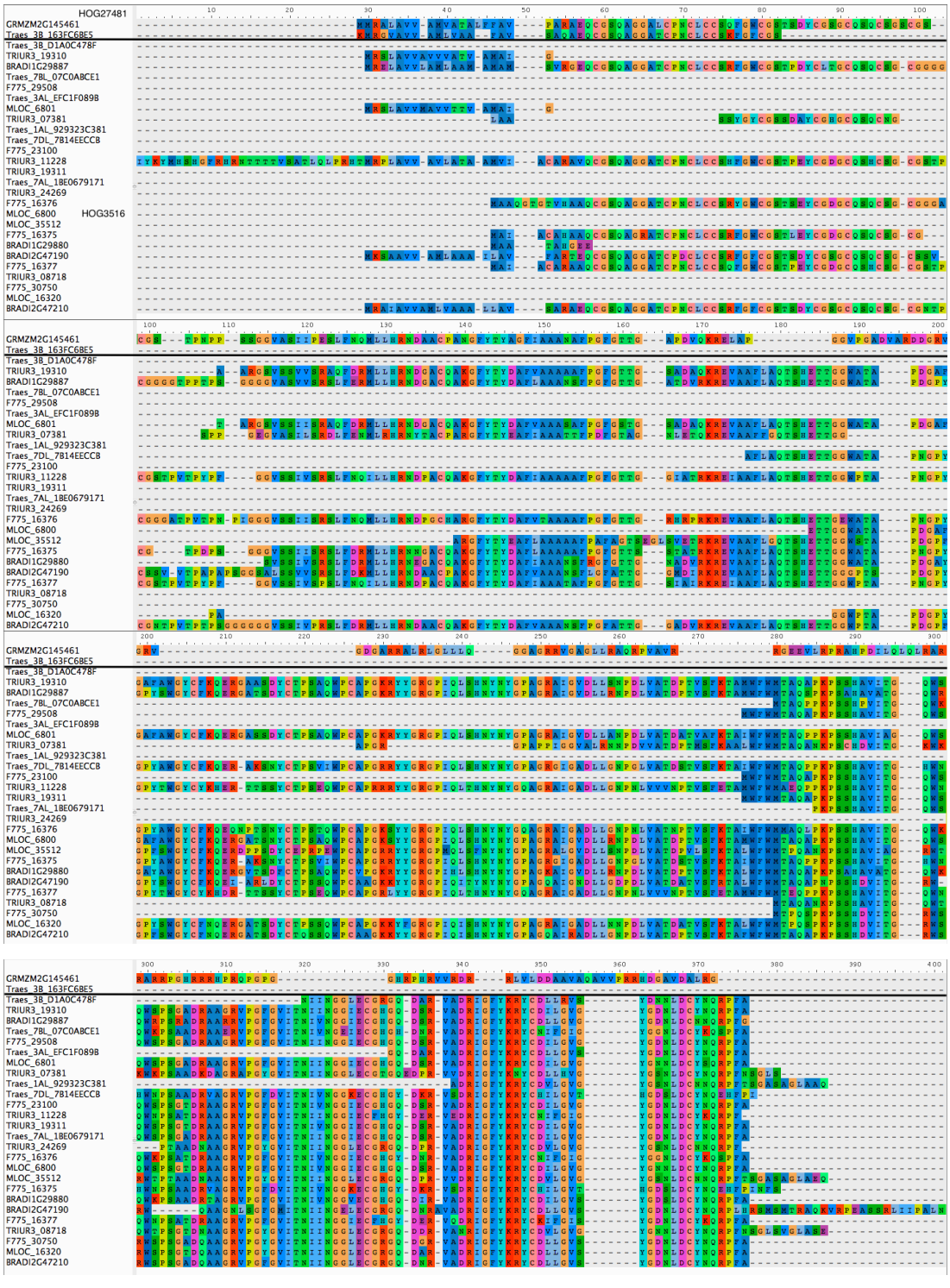


Figure B.2: Multiple sequence alignment of merged HOGs 22410 and 22409 (separated by black line).

Case under investigation: (Traes_3B_01F0A66DF, Traes_3B_E3185192A).

Alignment drawn with AliView (Larsson, 2014).



(figure continues on the next page)

(figure continues from the previous page)

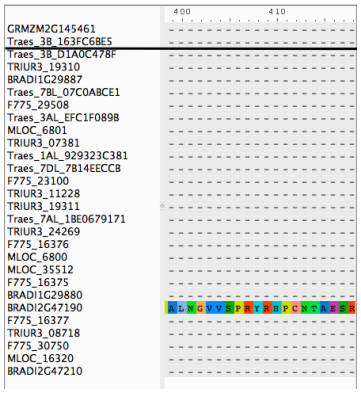
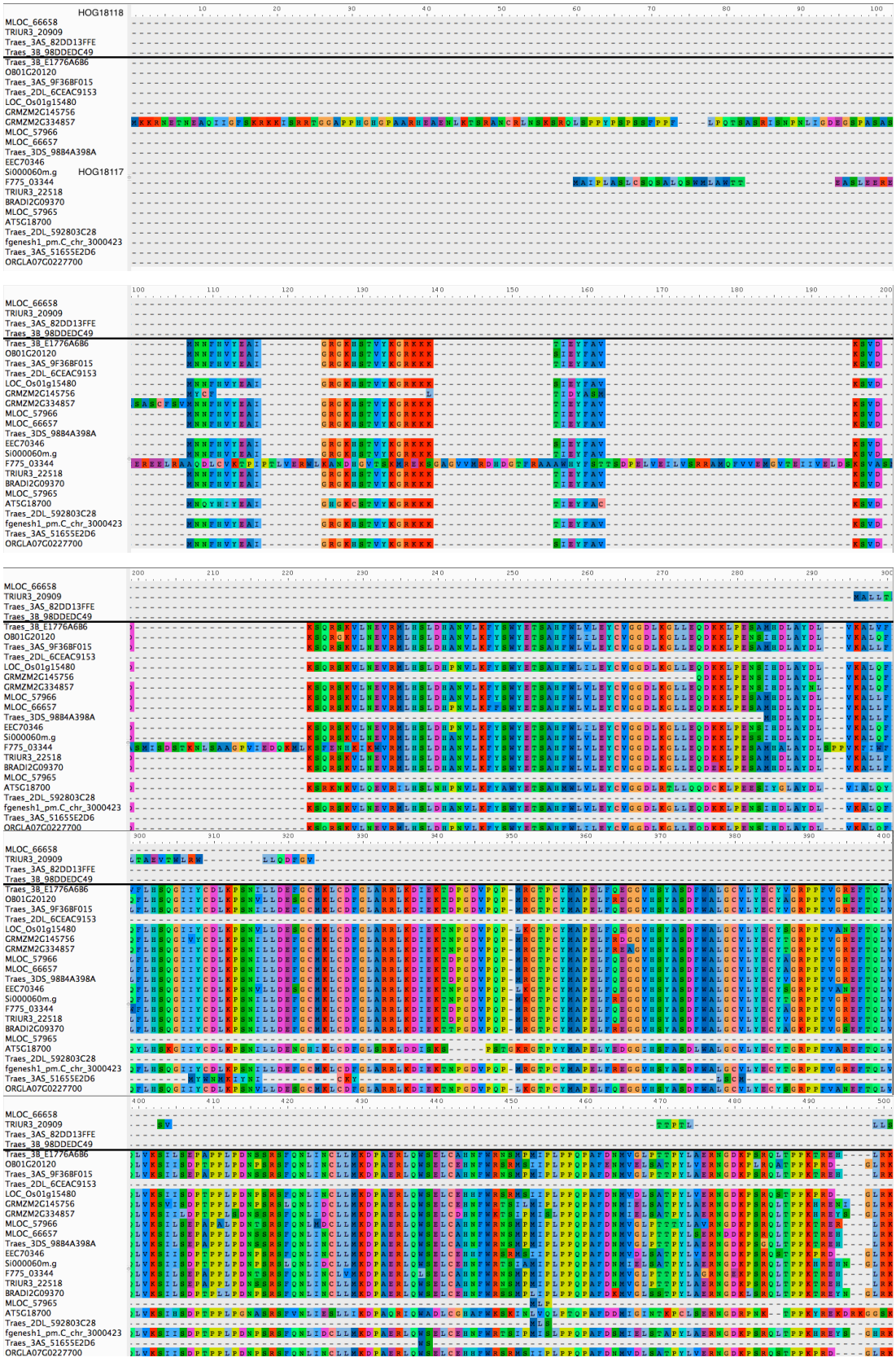


Figure B.3: Multiple sequence alignment of merged HOGs 27481 and 3516 (separated by black line).

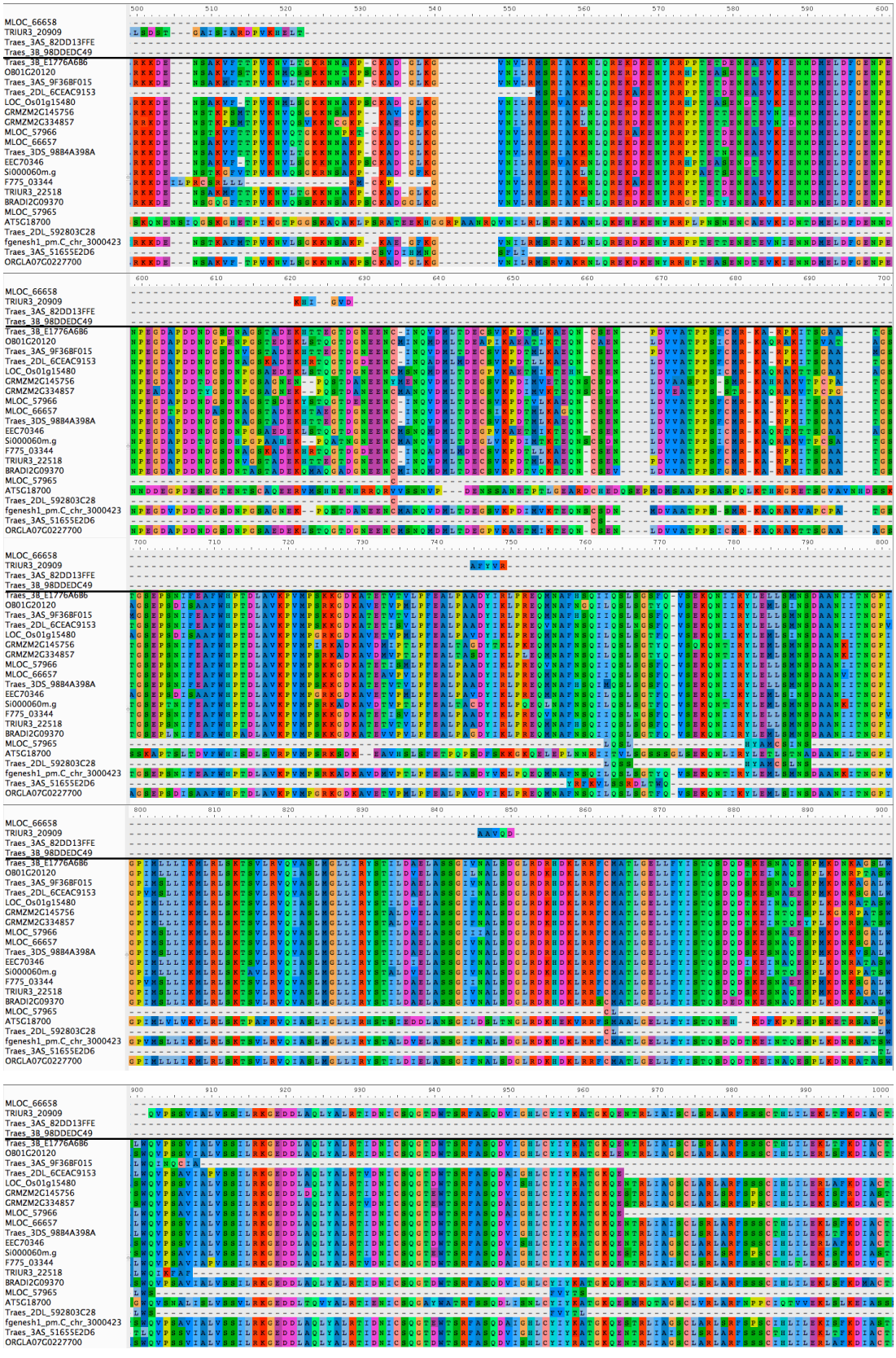
Case under investigation: (Traes_3B_163FC6BE5, Traes_3B_D1A0C478F).

Alignment drawn with AliView (Larsson, 2014).



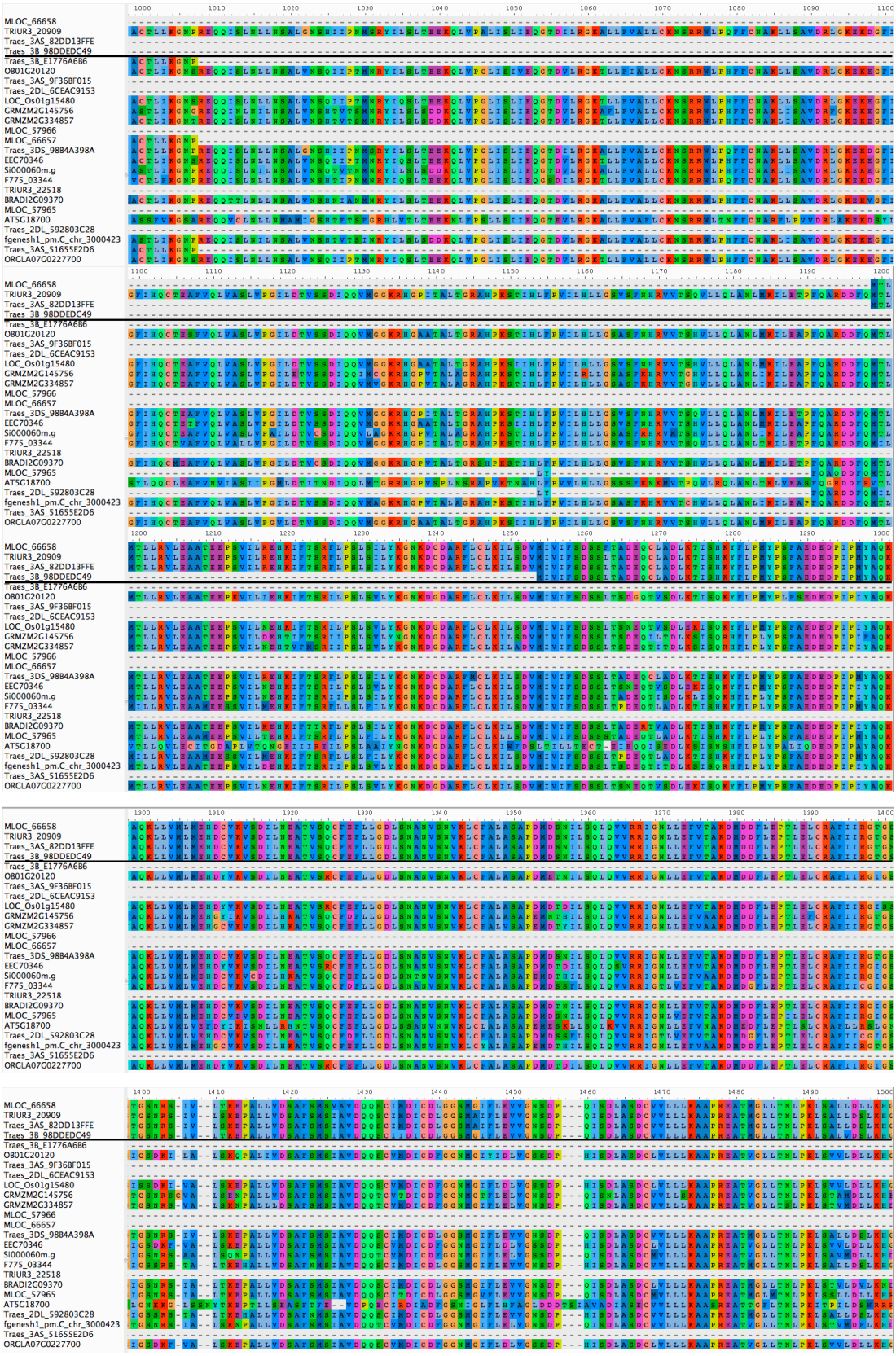
(figure continues on the next page)

(figure continues from the previous page)



(figure continues on the next page)

(figure continues from the previous page)



(figure continues on the next page)

(figure continues from the previous page)

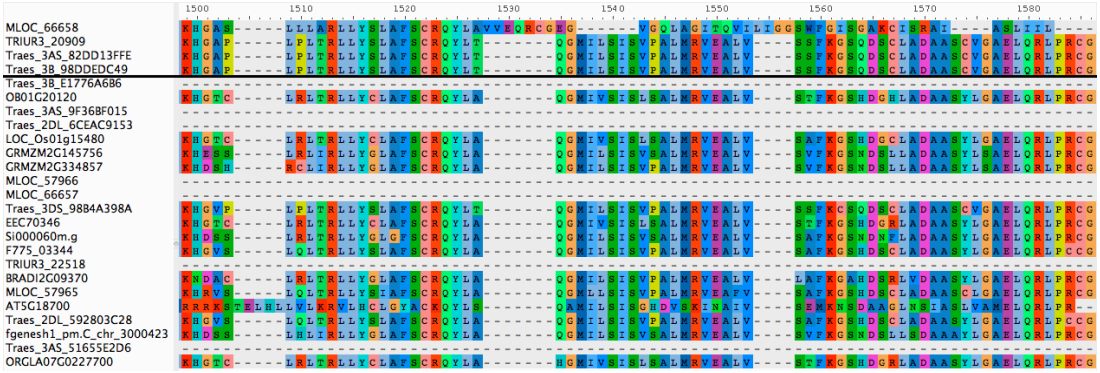
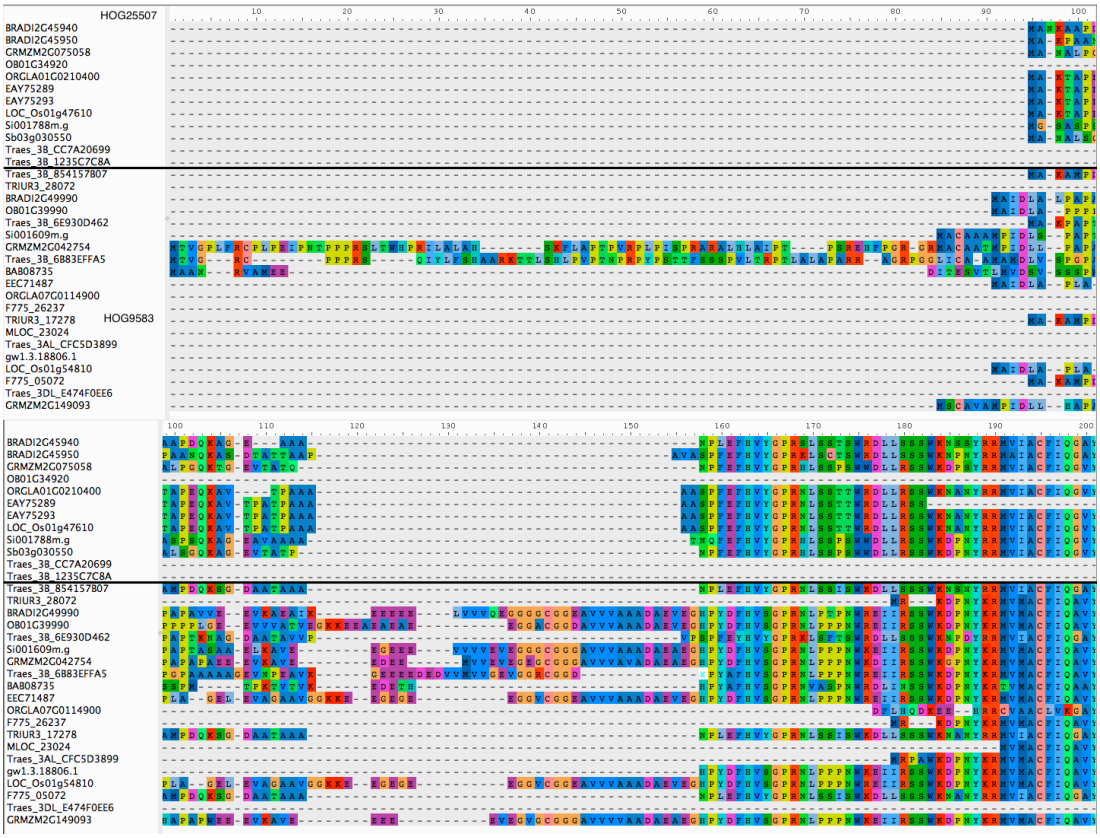


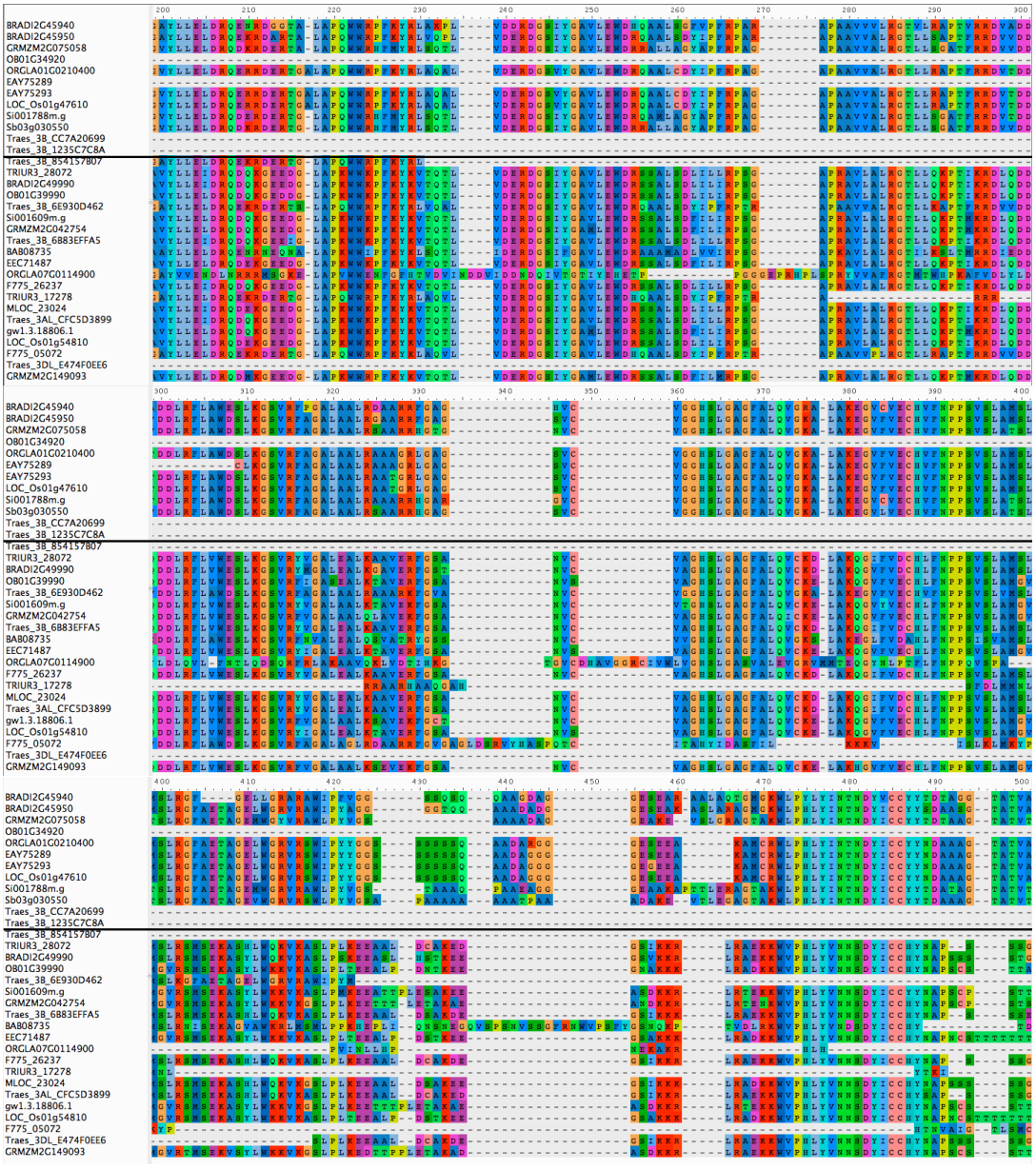
Figure B.4: Multiple sequence alignment of merged HOGs 18118 and 18117 (separated by black line).

Case under investigation: (Traes_3B_98DDEDC49, Traes_3B_E1776A6B6). Alignment drawn with AliView (Larsson, 2014).



(figure continues on the next page)

(figure continues from the previous page)



(figure continues on the next page)

(figure continues from the previous page)

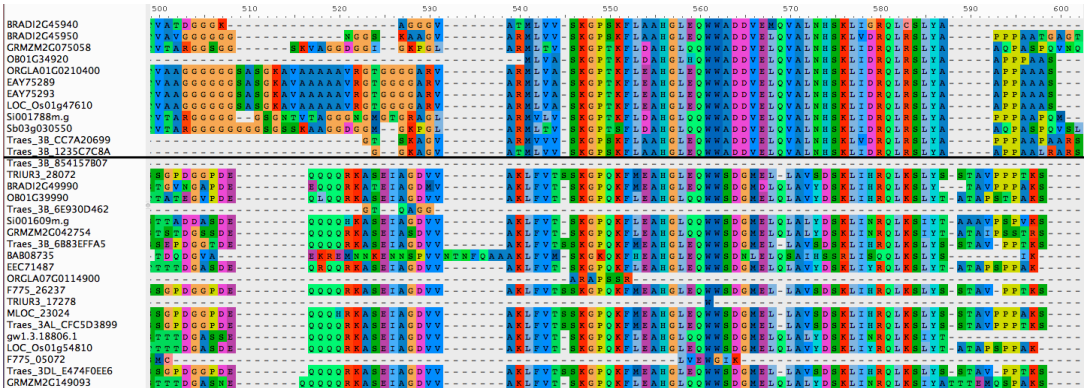


Figure B.5: Multiple sequence alignment of merged HOGs 25507 and 9583 (separated by black line).

Case under investigation: (Traes_3B_1235C7C8A, Traes_3B_854157B07). Alignment drawn with AliView (Larsson, 2014).

This case-by-case analysis with rather small HOGs indicates that bigger putative homologous families, especially those considering putative subsequence homology could improve performance of the heuristics already by providing them with more candidates. In addition, bigger putative families could perhaps contain enough information to make correct inference even when they included fragmented putative reference sequences.

B.6.4 Heuristic inference on previously discarded cases

To further explore the potential in merging putative homologous groups, and in general applying heuristics on pairs discarded earlier in the pipeline, we decided to apply our heuristics (collapsing with threshold 0.95, likelihood ratio heuristics with significance 0.01 and combination of the two) to the potentially fragmented cases identified with BLAST+ (Camacho *et al.*, 2009) less and more stringent mapping using input families calculated by GETHOGs (Altenhoff *et al.*, 2013) with default set of parameters on the input data listed in Table B.2.

As already elaborated, these previously not examined cases could be divided into 3 subgroups:

1) Not subjected to heuristics because at least one putative protein sequence was shorter than 50 AA
2) Not subjected to heuristics because putative protein sequences mutually overlap ≥10% of their sequence length in the MSA of corresponding putative protein family
3) Not subjected to heuristics because putative protein sequences were not assigned to the same putative protein family. There were also instances of short putative sequences and long overlap among them—we considered all of them for testing.

To obtain input putative families for heuristics where putative sequences were assigned to different HOGs, we merged corresponding HOGs, if possible. Surprisingly, only for a minority of cases we could find both putative sequences placed in a HOG (Tables B.15-B.16)—28/285 and 17/136 in BLAST+ (Camacho *et al.*, 2009) less and more stringently mapped cases respectively. This is due to GETHOGs algorithm (Altenhoff *et al.*, 2013) which discards putative sequences and relationships among putative sequences along its pipeline if they do not pass certain thresholds (in particular the length, pairwise score, number of putative orthologous relationships to merge putative subfamilies).

Both collapsing and likelihood ratio heuristics were usually able to identify more than 50% of split gene models on these previously discarded cases for which BLAST+ (Camacho *et al.*, 2009) mapping indicated fragmentation (Tables B.15-B.16).

As the number of cases subjected to heuristics was low, the fraction of fragmented cases confirmed by our tests might not be the best measure of heuristics' performance. Also, if putative homologous families are being merged, it might be better to examine their content first to avoid dubious mergings like in the case of (Traes_3B_01F0A66DF, Traes_3B_E3185192A) (Fig. B.2). Merging could be improved by requiring certain similarity levels or estimated evolutionary distances between putative homologous families. More than two families could also be merged into a single one.

Table B.15: Predictions on previously discarded cases identified with BLAST+ (Camacho *et al.*, 2009) less stringent mapping.

BLAST+ less stringent mapping					
	Short putative protein sequence(s) (7 cases)	**Long overlap (62 cases)**	**Different putative protein family (285; could test 28)**		
			Short putative protein sequence(s) (3 cases)	Long overlap (17 cases)	≥ 50AA, < 10% ovlp (8 cases)
Coll 0.95	6 (0.857)	46 (0.742)	1 (0.333)	9 (0.530)	3 (0.375)
LRH 0.01	7 (1.000)	61 (0.984)	2 (0.667)	12 (0.706)	8 (1.000)
Coll 0.95 + LRH 0.01	6 (0.857)	46 (0.742)	1 (0.333)	8 (0.471)	3 (0.375)

Table B.16: Predictions on previously dicarded cases identified with BLAST+ (Camacho *et al.*, 2009) more stringent mapping.

BLAST+ more stringent mapping					
	Short putative protein sequence(s) (3 cases)	**Long overlap** (15 cases)	**Different putative protein family (136; could test 17)**		
			Short putative protein sequence(s) (3 cases)	Long overlap (7 cases)	≥ 50AA, < 10% ovlp (7 cases)
Coll 0.95	3 (1.000)	11 (0.733)	1 (0.333)	4 (0.571)	3 (0.429)
LRH 0.01	3 (1.000)	15 (1.000)	2 (0.667)	5 (0.714)	7 (1.000)
Coll 0.95 + LRH 0.01	3 (1.000)	11 (0.733)	1 (0.333)	4 (0.571)	3 (0.429)

Table B.17: Number of predictions per putative wheat chromosome.

(table continues on the next page)

Input data		Predictions										
Putative chromosome	#gene models	#unambiguous (2 fragments per gene model)	#unambiguous (>2 fragments per gene model)	#total unambiguous			#total ambiguous			#total		
1AS	1,380	19	6	25			11			36		
1AL	2,464	15	0	15	40		4	15		19	55	
1BS	1,310	9	0	9			0			9		
1BL	2,578	18	2	20	29		7	7		27	36	
1DS	1,151	3	0	3			4			7		
1DL	2,268	20	9	29	32	101	2	6	28	31	38	129
2AS	2,084	21	0	21			0			21		
2AL	3,106	27	3	30	51		0	0		30	51	
2BS	2,237	12	0	12			0			12		
2BL	3,476	27	0	27	39		2	2		29	41	
2DS	2,153	42	9	51			21			72		
2DL	3,540	77	29	106	157	247	21	42	44	127	199	291
3AS	1,690	33	3	36			4			40		
3AL	2,534	49	18	67	103		3	7		70	110	
3B	5,609	58	0	58	58		8	8		66	66	
3DS	1,654	53	29	82			12			94		
3DL	2,342	26	3	29	111	272	2	14	29	31	125	301
4AS	1,499	12	0	12			4			16		
4AL	2,878	21	3	24	36		4	8		28	44	
4BS	1,697	6	0	6			4			10		
4BL	2,309	42	0	42	48		2	6		44	54	
4DS	1,655	13	0	13			10			23		
4DL	2,588	45	12	57	70	154	17	27	41	74	97	195

Table B.17: Number of predictions per putative wheat chromosome.

(table continues from the previous page)

Input data		Predictions											
Putative chromosome	#gene models	#unambiguous (2 fragments per gene model)	#unambiguous (>2 fragments per gene model)	#total unambiguous			#total ambiguous			#total		
5AS	939	12	3	15			0			15		
5AL	2,784	38	2	40	55		2	2		42	57	
5BS	1,094	11	0	11			4			15		
5BL	4,140	20	0	20	31		7	11		27	42	
5DS	1,063	14	0	14			0			14		
5DL	3,480	12	0	12	26	112	2	2	15	14	28	127
6AS	1,684	16	0	16			11			27		
6AL	1,989	31	3	34	50		0	11		34	61	
6BS	1,445	10	0	10			7			17		
6BL	1,925	21	0	21	31		8	15		29	46	
6DS	1,350	12	0	12			4			16		
6DL	2,186	28	0	28	40	121	4	8	34	32	48	155
7AS	2,320	48	9	57			15			72		
7AL	1,932	27	0	27	84		0	15		27	99	
7BS	1,485	17	0	17			5			22		
7BL	2,227	36	3	39	56		4	9		43	65	
7DS	2,335	28	9	37			4			41		
7DL	2,315	28	9	37	74	214	2	6	30	39	80	244
Total	90,895				**Total**	1,221		**Total**	221		**Total**	1,442

B.8 Exploring the FastTree parameters

Table B.18: Results of the simulations on the top-level HOGs (Altenhoff *et al.*, 2013) when FastTree default installation v2.1.7 (Price, Dehal and Arkin, 2010) was employed for tree reconstruction.

	`-pseudo`		`-mlacc 2 -slownni`		`-spr 4`		`-mlacc 2 -slownni -spr 4`		`-wag`		`-gamma`	
	#TP	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**
Collapsing												
0.65	31	7	30	6	30	7	30	6	30	8	30	7
0.75	36	10	35	10	36	10	35	10	35	11	36	10
0.85	52	17	53	18	53	18	53	18	49	19	53	18
0.9	66	23	67	23	67	24	67	23	66	24	67	24
0.95	78	28	80	29	78	29	80	29	78	30	78	29
LRH												
0.2	56	16	52	18	57	18	52	18	57	17	57	17
0.15	62	19	59	20	62	22	59	20	65	22	60	22
0.1	73	26	73	28	76	29	73	28	70	29	76	29
0.05	82	33	80	35	83	33	80	35	80	35	83	33
0.01	92	42	92	42	92	43	92	42	91	45	92	42
Combination (Coll 0.95 + LRH)												
0.2	47	13	46	15	49	15	46	15	48	15	49	14
0.15	51	14	50	16	51	17	50	16	55	18	50	17

(table continues on the next page)

(table continues from the previous page)

	-pseudo		-mlacc 2 -slownni		-spr 4		-mlacc 2 -slownni -spr 4		-wag		-gamma	
	#TP	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**
0.1	59	19	61	21	61	21	61	21	58	23	61	21
0.05	66	23	67	24	67	23	67	24	65	25	67	23
0.01	74	25	76	27	74	26	76	27	74	29	74	26

Table B.19: Results of the simulations on the top-level HOGs (Altenhoff *et al.*, 2013) when FastTree double-precision installation v2.1.10 (Price, Dehal and Arkin, 2010) was employed for tree reconstruction.

	default		-pseudo		-mlacc 2 -slownni		-spr 4		-mlacc 2 -slownni -spr 4		-pseudo -mlacc 2 -slownni -spr 4		-wag		-gamma		-lg	
	#TP	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**
Collapsing																		
0.65	35	7	36	7	36	7	35	7	36	7	36	7	37	6	35	7	36	8
0.75	43	10	43	10	43	11	44	10	44	11	44	11	43	10	43	10	45	11
0.85	61	18	62	18	60	20	61	18	60	20	61	20	57	21	61	18	57	18
0.9	78	21	78	22	76	22	78	21	76	22	76	22	73	24	78	21	71	25
0.95	84	29	84	29	85	29	84	29	85	29	85	29	84	30	84	29	87	30

(table continues on the next page)

(table continues from the previous page)

	default		-pseudo		-mlacc 2 -slownni		-spr 4		-mlacc 2 -slownni -spr 4		-pseudo -mlacc 2 -slownni -spr 4		-wag		-gamma		-lg	
	# TP	# FP	# TP	# FP	# TP	# FP	# TP	# FP	# TP	# FP	# TP	# FP	# TP	# FP	# TP	# FP	# TP	# FP
LRH																		
0.2	57	16	57	16	53	17	57	16	54	19	53	19	58	17	57	16	58	20
0.15	63	22	63	22	60	23	63	22	61	23	61	22	64	23	63	22	64	21
0.1	73	27	74	27	74	28	74	27	74	28	74	28	69	32	73	27	72	27
0.05	83	31	83	32	82	34	83	31	82	34	82	35	81	36	83	31	81	36
0.01	92	43	92	42	92	43	92	43	92	44	92	43	91	43	92	43	92	44
Combination (Coll 0.95 + LRH)																		
0.2	54	14	54	14	51	14	54	14	52	15	51	15	53	15	54	14	55	17
0.15	58	17	57	17	56	19	58	17	56	19	56	17	59	20	58	17	60	18
0.1	67	21	67	21	67	22	68	21	67	22	67	22	62	24	67	21	68	23
0.05	74	23	74	24	74	24	74	23	74	24	74	25	72	27	74	23	74	27
0.01	80	27	80	26	81	27	80	27	81	27	81	27	79	29	80	27	84	29

B.9 Exploring the RAxML parameters

Table B.20: Results of the simulations on the top-level HOGs (Altenhoff *et al.*, 2013) when RAxML v8.2.12 (Alexandros Stamatakis, 2014) was employed for tree reconstruction.

	`-m PROTCATJTT -c 20`		`-m PROTCATLG -c 20`		`-m PROTGAMMAAUTO`		`-m PROTGAMMAGTR`	
	#TP	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**
Collapsing								
0.65	26	5	27	5	21	6	24	6
0.75	33	6	35	9	29	8	32	8
0.85	50	14	50	17	50	15	55	12
0.9	67	18	64	22	67	22	64	17
0.95	77	28	78	26	74	29	74	23
LRH								
0.2	49	16	45	16	45	16	49	16
0.15	53	17	56	17	52	17	57	17
0.1	65	20	67	20	57	19	64	21
0.05	79	28	77	26	74	26	73	26
0.01	82	34	83	36	80	33	81	34
Combination (Coll 0.95 + LRH)								
0.2	46	15	44	15	42	15	47	13
0.15	50	16	53	16	49	16	55	14
0.1	58	19	60	18	53	18	60	17

(table continues on the next page)

(table continues from the previous page)

	`-m PROTCATJTT -c 20`		`-m PROTCATLG -c 20`		`-m PROTGAMMAAUTO`		`-m PROTGAMMAGTR`	
	#TP	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**	**#TP**	**#FP**
0.05	68	24	68	22	65	22	67	20
0.01	70	26	72	25	69	28	73	22
#pairs tested*	94	99	94	99	91	94	90	98

* Some pairs had to be excluded from the analysis either because RAxML requires at least four putative sequences in a dataset or because RAxML ran into dataset-dependent numerical problems during optimization and crashed.

B.10 Preliminary investigation of empirical distributions

Thorough investigation of empirical distributions of the likelihood ratio value could reveal insights on the predictive power of the likelihood ratio heuristic, its risk of misclassification and potential ways of parametrising the distribution or normalizing the likelihood ratio values to make them comparable across examined cases. Being out of the scope of the project, we only investigated a few random cases to see if there was maybe a self-revealing pattern.

We investigated 12 randomly chosen empirical distributions of the likelihood ratio value:

- 6 distributions from experiment on 100 artificially fragmented putative protein sequences, HOGs input data (Altenhoff *et al.*, 2013), FastTree v2.1.8 (Price, Dehal and Arkin, 2010) tree reconstruction with default settings (described in section 3.2.7)
    - 3 distributions where the heuristic made correct prediction
    - 3 distributions where the heuristic made wrong prediction

In the rest of the section, we sometimes refer to them as *same gene cases*.

- 6 distributions from experiment on 100 candidate pairs derived by artificial fragmentation of putative paralogous protein sequences, HOGs data, FastTree tree reconstruction with default settings (described in section 3.2.8)
    - 3 distributions where the heuristic made correct prediction
    - 3 distributions where the heuristic made wrong prediction

  In the rest of the section, we sometimes refer to them as *putative paralogous cases*.

No two cases were from the same HOG, i.e. all cases were from different input putative protein families. Files with likelihood ratio values for selected cases are available at https://doi.org/10.6084/m9.figshare.11733975.v1.

We plotted each of the empirical distributions (with corresponding likelihood ratio value) (Fig. B.6), their histograms (Fig. B.7-B.8), checked measures of centrality and dispersion (Table B.21), and tried to fit four distributions: χ^2 (Abbe, 1863), gamma (Laplace, 1836), Weibull (Weibull, 1951) and log-normal (Weber, 1834) to the obtained empirical values greater than zero (Table B.22-B.25).

Looking at the descriptive statistics (Table B.21), it seems that mean and median of the empirical distributions for putative paralogous cases were higher than for the same gene cases. This could be due to the modelling under the assumption that does not hold (fragments of the same gene model) but actually, it is indeed what we expect to see even under correct working assumption—because for fragments derived from paralogs, the ratio of likelihoods should be greater than for fragments derived from the same gene. The same argument is valid for the spread of distributions, i.e. explains why we expected to see and observed here distributions for putative paralogous cases having wider range than those for split gene cases (Table B.21, Fig. B.6-B.8). However, we have to keep in mind that for example close paralogs can yield distributions similar to the ones of split genes or inadequate modelling can lead to high likelihood ratios for truly fragmented

cases. As for the negative values of likelihood ratio, these were bootstrap samples for which reconstructed trees were suboptimal. All selected empirical distributions were unimodal and positively (right) skewed. There were outliers in both same gene and putative paralogous cases distributions regardless of the accuracy of the heuristic outcome. This could be attributed to the (software) modelling, perhaps getting stuck in a local optimum.

At first glance, four density curves strike from Figure B.6: f_4 from panel a) and f_1 , f_4 and f_5 from panel b), only b) f_1 depicting correct prediction. In all four cases, a large proportion of the data was within relatively small range right of the 0. This could indicate higher risk of misclassification. Although such distributions could be generated by fragments of the same gene model (f_4 on panel a)), they could also appear due to close paralogous relationships.

Overall, we have too few cases to make any general conclusions but there might be already two directions for future research. First, it might be worth comparing more exhaustively the spread of empirical distributions—for split gene cases verus paralogous gene cases as there are indications that larger range could be due to paralogy. Second, investigating quantiles of the empirical distributions as they could indicate risk of misclassification.

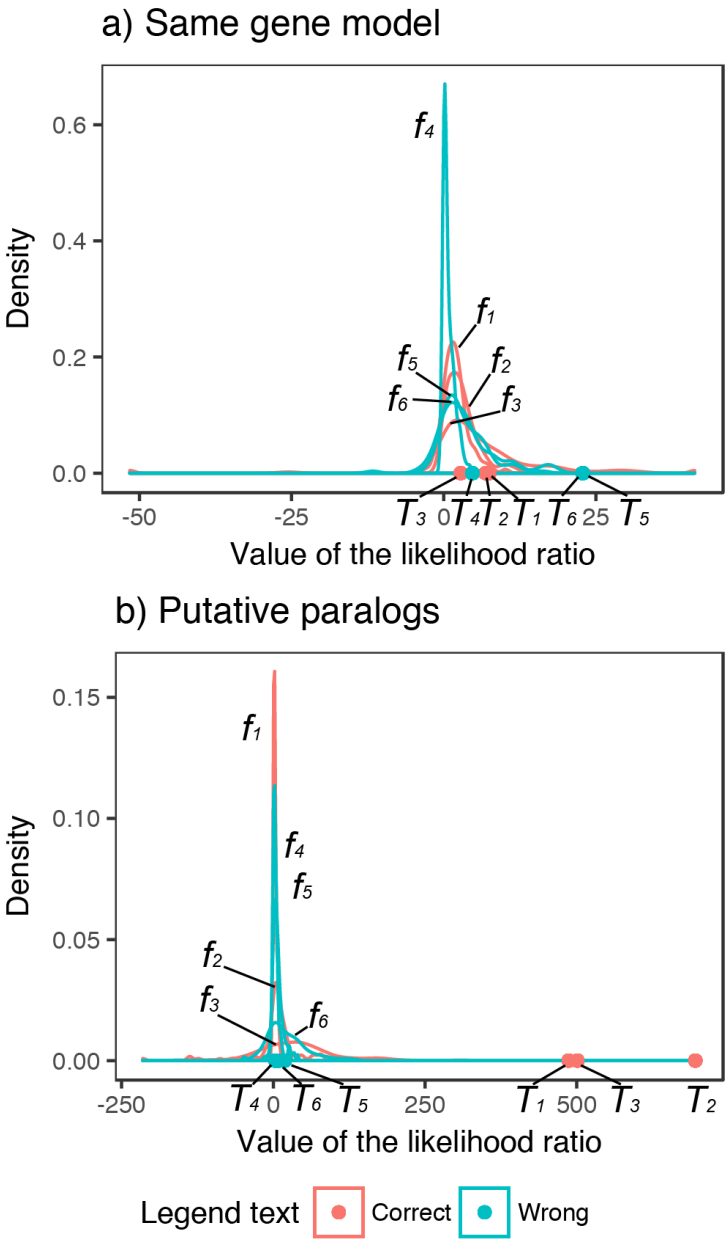


Figure B.6: Randomly chosen empirical distributions of the likelihood ratio value and corresponding values of the likelihood ratio.

In a) Same gene model, candidate input fragments were derived from a putative protein sequence of a single gene model while in b) Putative paralogs, we artificially fragmented pairs of putative protein sequences of putative paralogous gene models and subjected them to the heuristic. Distributions and *T* values in red depict cases where the likelihood ratio heuristic made correct inference while the blue cases depict cases of wrong inference.

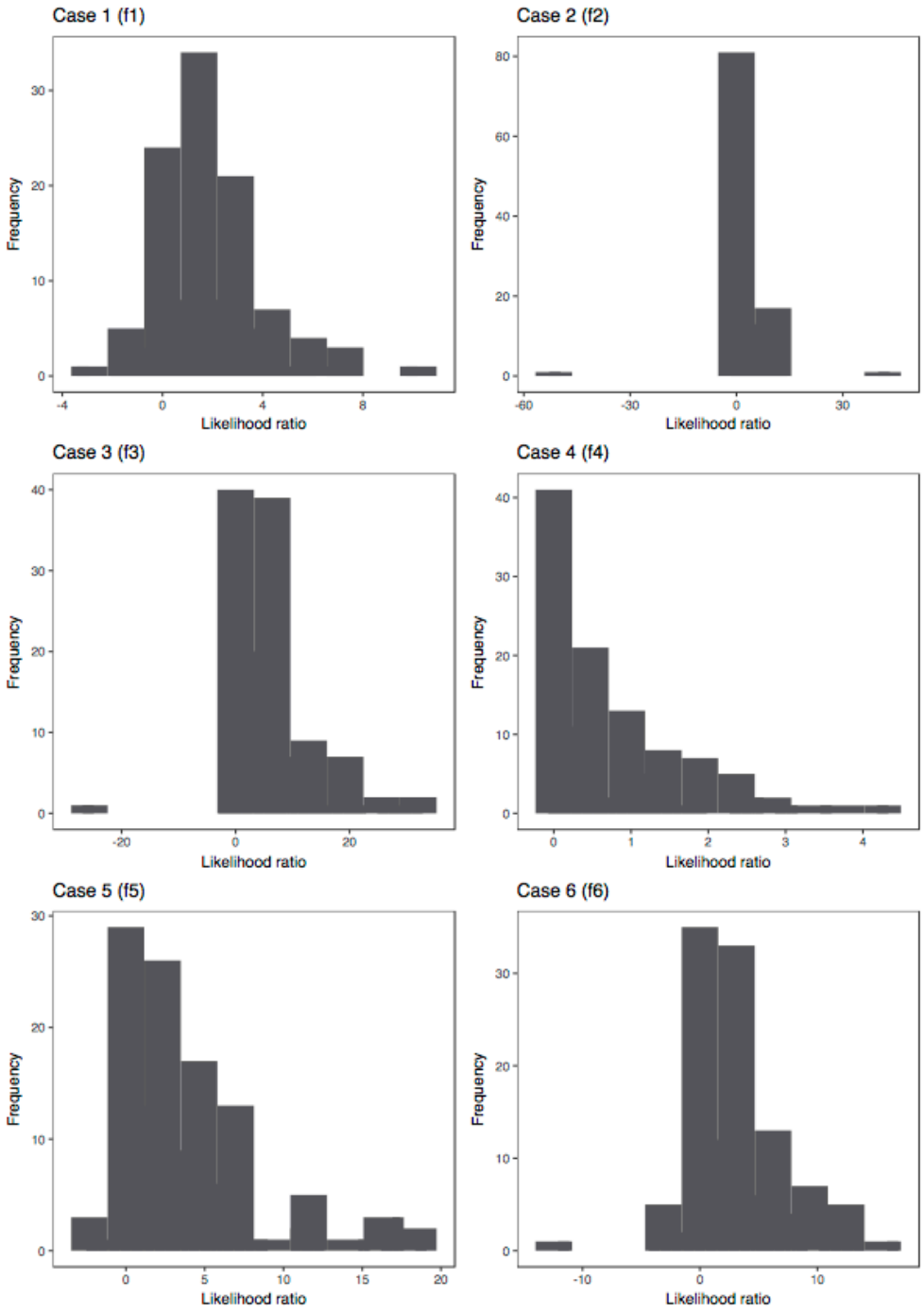


Figure B.7: Histograms of empirical (bootstrap) likelihood ratio values for randomly chosen cases derived from a putative protein sequence of the same gene model (same cases as in Fig. B.6a).

In cases 1-3, the likelihood ratio heuristic made correct inference while for cases 4-6 it was wrong.

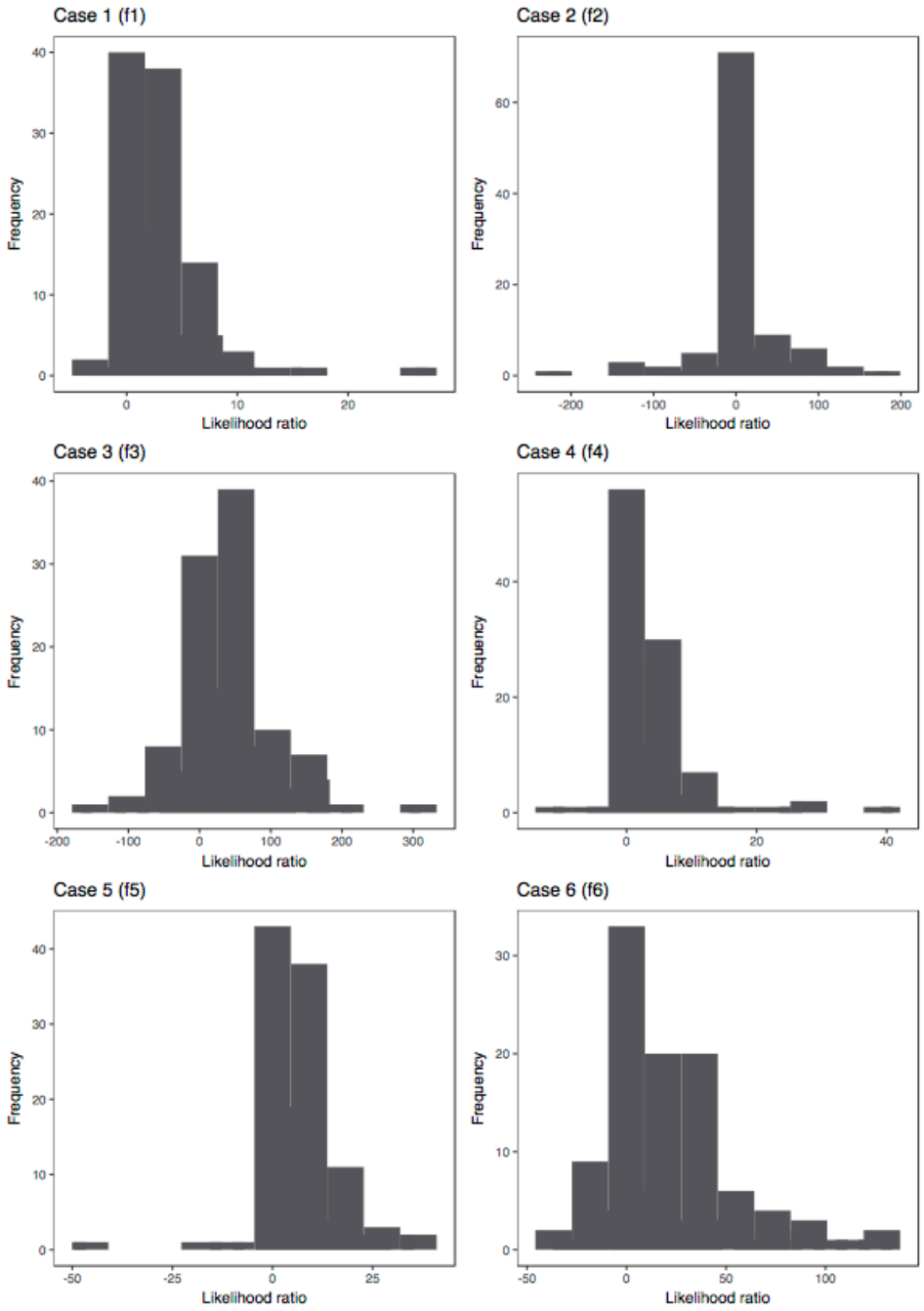


Figure B.8: Histograms of empirical (bootstrap) likelihood ratio values for randomly chosen cases derived from pairs of putative protein sequences of putative paralogous gene models (same cases as in Fig. B.6b).

In cases 1-3, the likelihood ratio heuristic made correct inference while for cases 4-6 it was wrong.

Table B.21: Descriptive statistics for randomly selected cases (same cases as in Fig. B.6-B.8).

	Centrality		Dispersion	
	Mean	**Median**	**Range**	**IQR**
Same gene model				
Correct				
Case 1	1.896	1.673	13.128	2.237
Case 2	2.898	2.489	92.82	3.192
Case 3	6.029	4.210	57.72	7.262
Wrong				
Case 4	0.7696	0.3340	4.256	1.124
Case 5	3.9110	2.4070	20.878	4.8425
Case 6	3.0670	2.4250	28	4.1675
Putative paralogous gene models				
Correct				
Case 1	3.055	2.146	29.658	3.112
Case 2	6.797	6.608	398	16.612
Case 3	38.8100	35.8400	461.2	68.5735
Wrong				
Case 4	3.894	2.068	50.48	4.494
Case 5	6.000	4.907	81.96	8.407
Case 6	21.9600	15.2100	164.99	38.3205

We also tried to fit four distributions to the selected empirical data: χ^2 (Abbe, 1863), gamma (Laplace, 1836), Weibull (Weibull, 1951) and log-normal (Weber, 1834) (Table B.22-B.25). No obvious parameterisation emerged as a unique solution and this might be a challenging problem to tackle.

Table B.22: Fitting distributions—cases derived from the same gene model and correctly inferred as fragments (same data as in Fig. B.6-B.7, Table B.21).

Four distributions (χ^2, gamma, Weibull and log-normal) were fitted to each case using R library fitdistrplus (Delignette-Muller, Dutang and Others, 2015; R Core Team, 2017) followed by Kolmogorov-Smirnov goodness-of-fit test (Kolmogorov, 1933; Smirnov, 1948).

	χ^2 Estimated degrees of freedom (Estimated standard error)	**gamma** Estimated shape parameter (Estimated standard error) Estimated rate parameter (Estimated standard error)	**Weibull** Estimated shape parameter (Estimated standard error) Estimated scale parameter (Estimated standard error)	**log-normal** Estimated μ (Estimated standard error) Estimated σ (Estimated standard error)
	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value
Case 1	2.429 (0.192)	1.335 (0.183) 0.576 (0.095)	1.225 (0.103) 2.473 (0.227)	0.422 (0.122) 1.136 (0.086)
	0.670	0.671	0.871	0.060

(table continues on the next page)

(table continues from the previous page)

	χ^2 Estimated degrees of freedom (Estimated standard error)	**gamma** Estimated shape parameter (Estimated standard error) Estimated rate parameter (Estimated standard error)	**Weibull** Estimated shape parameter (Estimated standard error) Estimated scale parameter (Estimated standard error)	**log-normal** Estimated μ (Estimated standard error) Estimated σ (Estimated standard error)
	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value
Case 2	2.836 (0.201)	0.966 (0.121) 0.277 (0.045)	0.947 (0.070) 3.399 (0.382)	0.650 (0.131) 1.292 (0.092)
	0.589	0.660	0.458	0.032
Case 3	4.584 (0.278)	0.960 (0.122) 0.145 (0.024)	0.967 (0.077) 6.537 (0.731)	1.289 (0.130) 1.269 (0.092)
	< 2.2e-16	0.990	0.997	0.389

Table B.23: Fitting distributions—cases derived from the same gene model but inferred as fragments of putative paralogous gene models (same data as in Fig. B.6-B.7, Table B.21).

Four distributions (χ^2, gamma, Weibull and log-normal) were fitted to each case using R library fitdistrplus (Delignette-Muller, Dutang and Others, 2015; R Core Team, 2017) followed by Kolmogorov-Smirnov goodness-of-fit test (Kolmogorov, 1933; Smirnov, 1948).

	χ^2	**gamma**	**Weibull**	**log-normal**
	Estimated degrees of freedom (Estimated standard error)	Estimated shape parameter (Estimated standard error) Estimated rate parameter (Estimated standard error)	Estimated shape parameter (Estimated standard error) Estimated scale parameter (Estimated standard error)	Estimated μ (Estimated standard error) Estimated σ (Estimated standard error)
	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value
Case 4	0.946 (0.087)	0.534 (0.064) 0.673 (0.123)	0.661 (0.055) 0.616 (0.099)	-1.409 (0.207) 2.037 (0.146)
	0.945	0.813	0.635	0.143
Case 5	3.177 (0.223)	0.901 (0.115) 0.210 (0.035)	0.940 (0.077) 4.180 (0.485)	0.810 (0.143) 1.381 (0.101)
	0.019	0.997	0.998	0.2814

(table continues on the next page)

(table continues from the previous page)

	χ^2 Estimated degrees of freedom (Estimated standard error)	**gamma** Estimated shape parameter (Estimated standard error) Estimated rate parameter (Estimated standard error)	**Weibull** Estimated shape parameter (Estimated standard error) Estimated scale parameter (Estimated standard error)	**log-normal** Estimated μ (Estimated standard error) Estimated σ (Estimated standard error)
	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value
Case 6	3.205 (0.237)	0.980 (0.134) 0.239 (0.042)	1.040 (0.093) 4.162 (0.459)	0.822 (0.167) 1.525 (0.118)
	0.100	0.671	0.876	0.045

Table B.24: Fitting distributions—cases derived from putative paralogous gene models and correctly inferred as such (same data as in Fig. B.6, B.8, Table B.21).

Four distributions (χ^2, gamma, Weibull and log-normal) were fitted to each case using R library fitdistrplus (Delignette-Muller, Dutang and Others, 2015; R Core Team, 2017) followed by Kolmogorov-Smirnov goodness-of-fit test (Kolmogorov, 1933; Smirnov, 1948).

	χ^2 Estimated degrees of freedom (Estimated standard error)	**gamma** Estimated shape parameter (Estimated standard error) Estimated rate parameter (Estimated standard error)	**Weibull** Estimated shape parameter (Estimated standard error) Estimated scale parameter (Estimated standard error)	**log-normal** Estimated μ (Estimated standard error) Estimated σ (Estimated standard error)
	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value
Case 1	2.948 (0.213)	1.099 (0.144) 0.323 (0.053)	1.034 (0.080) 3.453 (0.367)	0.705 (0.127) 1.214 (0.090)
	0.698	0.808	0.674	0.390
Case 2	10.788 (0.505)	0.719 (0.099) 0.032 (0.006)	0.780 (0.066) 19.166 (2.969)	2.283 (0.154) 1.349 (0.109)
	8.66e-05	0.103	0.273	0.937

(table continues on the next page)

(table continues from the previous page)

	χ^2 Estimated degrees of freedom (Estimated standard error)	**gamma** Estimated shape parameter (Estimated standard error) Estimated rate parameter (Estimated standard error)	**Weibull** Estimated shape parameter (Estimated standard error) Estimated scale parameter (Estimated standard error)	**log-normal** Estimated μ (Estimated standard error) Estimated σ (Estimated standard error)
	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value
Case 3	42.713 (1.062)	1.347 (0.200) 0.021 (0.004)	1.193 (0.106) 67.158 (6.894)	3.731 (0.123) 1.055 (0.087)
	4.792e-09	0.399	0.450	0.039

Table B.25: Fitting distributions—cases derived from putative paralogous gene models and incorrectly inferred as fragments of the same gene model (same data as in Fig. B.6, B.8, Table B.21).

Four distributions (χ^2, gamma, Weibull and log-normal) were fitted to each case using R library fitdistrplus (Delignette-Muller, Dutang and Others, 2015; R Core Team, 2017) followed by Kolmogorov-Smirnov goodness-of-fit test (Kolmogorov, 1933; Smirnov, 1948).

	χ^2 Estimated degrees of freedom (Estimated standard error)	**gamma** Estimated shape parameter (Estimated standard error) Estimated rate parameter (Estimated standard error)	**Weibull** Estimated shape parameter (Estimated standard error) Estimated scale parameter (Estimated standard error)	**log-normal** Estimated μ (Estimated standard error) Estimated σ (Estimated standard error)
	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value
Case 4	3.113 (0.226)	0.769 (0.100) 0.162 (0.029)	0.831 (0.067) 4.261 (0.577)	0.780 (0.155) 1.458 (0.110)
	0.031	0.401	0.577	0.245
Case 5	5.634 (0.327)	1.111 (0.149) 0.142 (0.024)	1.065 (0.088) 7.996 (0.843)	1.541 (0.126) 1.183 (0.089)
	0.009	0.995	0.998	0.249

(table continues on the next page)

(table continues from the previous page)

	χ^2 Estimated degrees of freedom (Estimated standard error)	**gamma** Estimated shape parameter (Estimated standard error) Estimated rate parameter (Estimated standard error)	**Weibull** Estimated shape parameter (Estimated standard error) Estimated scale parameter (Estimated standard error)	**log-normal** Estimated μ (Estimated standard error) Estimated σ (Estimated standard error)
	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value	K-S test *p*-value
Case 6	16.538 (0.632)	0.845 (0.117) 0.0271 (0.005)	0.921 (0.085) 30.161 (3.891)	2.744 (0.167) 1.466 (0.117)
	8.133e-13	0.085	0.521	0.071

B.11 Double likelihood ratio heuristic

Let the heuristic hypotheses be:

Heuristic 1:	Heuristic 2:
H_0: H_s	H_0: H_p
H_1: H_p	H_1: H_s

with the likelihood ratio value, in each heuristic, defined as $T = 2ln\frac{L(H_1)}{L(H_0)}$, *L()* denoting the maximum estimator under each hypothesis.

Bootstrap sampling in Heuristic 1 could be kept as it is while in Heuristic 2 it could be performed in two steps:

1. Break the multiple sequence alignment into two parts, let's call them *left* (first n_1 residues) and *right* (remaining n_2 residues) for simplicity, each spanning one candidate fragment.
2. Derive first n_1 residues of a sample from the left and the remaining n_2 residues of a sample from the right. Concatenate the subsamples.

There are four possible outcomes of these two heuristics:

1. H_0 selected by Heuristic 1, H_1 selected by Heuristic 2. We suggest this indicates that fragments were derived from the same gene.
2. H_1 selected by Heuristic 1, H_0 selected by Heuristic 2. We suggest this indicates that fragments were derived from paralogous genes.
3. H_0 selected by both heuristics. No discrimination between the models is possible which could happen if fragments were derived from very close paralogs or more generally, if there is too little information at hand to discriminate between the two hypotheses.
4. H_1 selected by both heuristics. This indicates that neither model is appropriate. It could happen if the evolutionary process is not tree-like (e.g. if there is recombination or gene fusion) or if some of the assumptions of the evolutionary model are violated. Furthermore, it could be due to the data quality, tools and their settings used in the pipeline or perhaps the putative sequences are indeed fully assembled and happen to be that short due to deletion.

However, we are aware that, again, in at least one of the heuristics we suggest working under the assumption which is not satisfied, i.e. performing the Heuristic 1 for fragments derived from paralogs, performing the Heuristic 2 when fragments were indeed derived from the same gene (and even performing both heuristics when sequences were derived from unrelated genes but the homology inference method misclassified them into the same putative gene (protein) family).

B.12 Investigation of likelihoods of reconstructed trees with and without using input topology

When reconstructing a tree under the H_p, we performed two tree searches—one without an input topology and one providing an input topology, as explained in section 3.2.4. To gain better understanding of its effects, we chose six experiments and investigated likelihoods of obtained trees and their ratios.

Chosen experiments:

- 100 artificially fragmented putative protein sequences, Ensembl putative protein families (Vilella *et al.*, 2009; Cunningham *et al.*, 2019; Howe *et al.*, 2020), FastTree v2.1.8 (Price, Dehal and Arkin, 2010) reconstruction with default settings (described in section 3.2.7)
- 100 artificially fragmented putative protein sequences, HOGs (Altenhoff *et al.*, 2013), FastTree v2.1.8 reconstruction with default settings (described in section 3.2.7)
- 100 artificially fragmented putative protein sequences, HOGs, RAxML v8.2.12 (Alexandros Stamatakis, 2014) reconstruction with settings equivalent to FastTree default settings (described in section 3.2.13)
- 100 candidate pairs derived by artificial fragmentation of putative paralogous protein sequences, Ensembl putative protein families, FastTree reconstruction with default settings (described in section 3.2.8)

- 100 candidate pairs derived by artificial fragmentation of putative paralogous protein sequences, HOGs, FastTree reconstruction with default settings (described in section 3.2.8)
- 100 candidate pairs derived by artificial fragmentation of putative paralogous protein sequences, HOGs, RAxML v8.2.12 reconstruction with settings equivalent to FastTree default settings (described in section 3.2.13)

Likelihoods of all reconstructed trees can be downloaded from https://doi.org/10.6084/m9.figshare.11733672.v1.

For each of the experiments above, we:

1. Counted the number of times when a reconstructed tree without providing an input topology had higher likelihood than a tree reconstructed considering an input topology, the number of times when a reconstructed tree without providing an input topology had equal likelihood as a tree reconstructed considering an input topology, and the number of times when a reconstructed tree using an input topology had higher likelihood than a tree reconstructed with no input topology. As can be seen in Table B.26, in two experiments a tree search with an input topology reported a better tree for the majority of examined cases while in every experiment it was helpful for at least ~¼ of the cases under investigation. This suggests that considering an input topology can be beneficial.
2. Plotted a histogram of the differences between the likelihoods:
   likelihood of the ML tree starting from an input topology -
   likelihood of the ML tree without an input topology.
   As can be seen from Figure B.9 and as expected given the counts in Table B.26, the differences were usually ≥ 0. The differences changed depending on a dataset (Fig. B.9a v Fig. B.9c, Fig. B.9b v Fig. B.9d) and a tree building tool (Fig. B.9c v Fig. B.9e, Fig. B.9d v Fig. B.9f). This could indicate the need to choose a tool and adjust its parameters to the dataset under examination. However, given the

outcomes of our investigation on different phylogenetic tools and their parameters (sections 3.2.13, 3.3.7), it could also be that this double tree search helps to even out downsides of the chosen phylogenetic tool and its parameters. Further analysis could provide more insights.

3. To see how this more exhaustive search affects the ratio of likelihoods, we plotted the distribution of ratios as it is in the current pipeline (taking the higher value) and the distribution of ratios considering only a tree search without an input topology (Fig. B.10). Considering an input topology had the greatest effect when the trees were reconstructed for Ensembl putative protein families by FastTree (Fig. B.10a-b). The effect was lower when FastTree was applied to HOGs (Fig. B.10c-d) indicating that default FastTree parameters might be more suitable for HOGs than for Ensembl families, and that the parameters should be investigated in order to improve tree search (for both Ensembl and HOGs data but for Ensembl data in particular). Considering an input topology had the lowest effect when trees were reconstructed from HOGs using RAxML (Fig. B.10e-f). That was expected since the RAxML tree search is more exhaustive than the tree search by FastTree. Overall, this emphasises the importance of exhaustive tree search and the need for choosing appropriate tree building methods and their parameters. Similarly as above under point 2., it could be that the double tree search mitigates shortcomings of tree-reconstruction algorithm to the extent meaningful for the likelihood ratio heuristic.

Taking into account all outcomes of this analysis, we would advise performing tree search under the H_p with and without including a starting topology. It could be beneficial for at least part of the cases under investigation. Furthermore, maybe it is the key of the robustness of the likelihood ratio heuristic observed in the experiments described in section 3.2.13 (results in section 3.3.7).

Table B.26: Counting the number of times when each of the tree searches under the H_p (with an input topology, without an input topology) found a more optimal tree.

	100 (FastTree) or 94 (RAxML) artificially fragmented putative protein sequences*			100 (FastTree) or 99 (RAxML) pairs from artificially fragmented putative paralogous protein sequences*		
	W INPUT	**=**	**W/O INPUT**	**W INPUT**	**=**	**W/O INPUT**
FastTree, default parameters (Ensembl)	57	6	37	55	3	42
FastTree, default parameters (HOGs)	29	33	38	49	12	39
RAxML, parameters as in FastTree default (HOGs)	45	27	22	24	17	58

* Some pairs had to be excluded from the analysis either because RAxML requires at least four putative sequences in a dataset or because RAxML ran into dataset-dependent numerical problems during optimization and crashed.

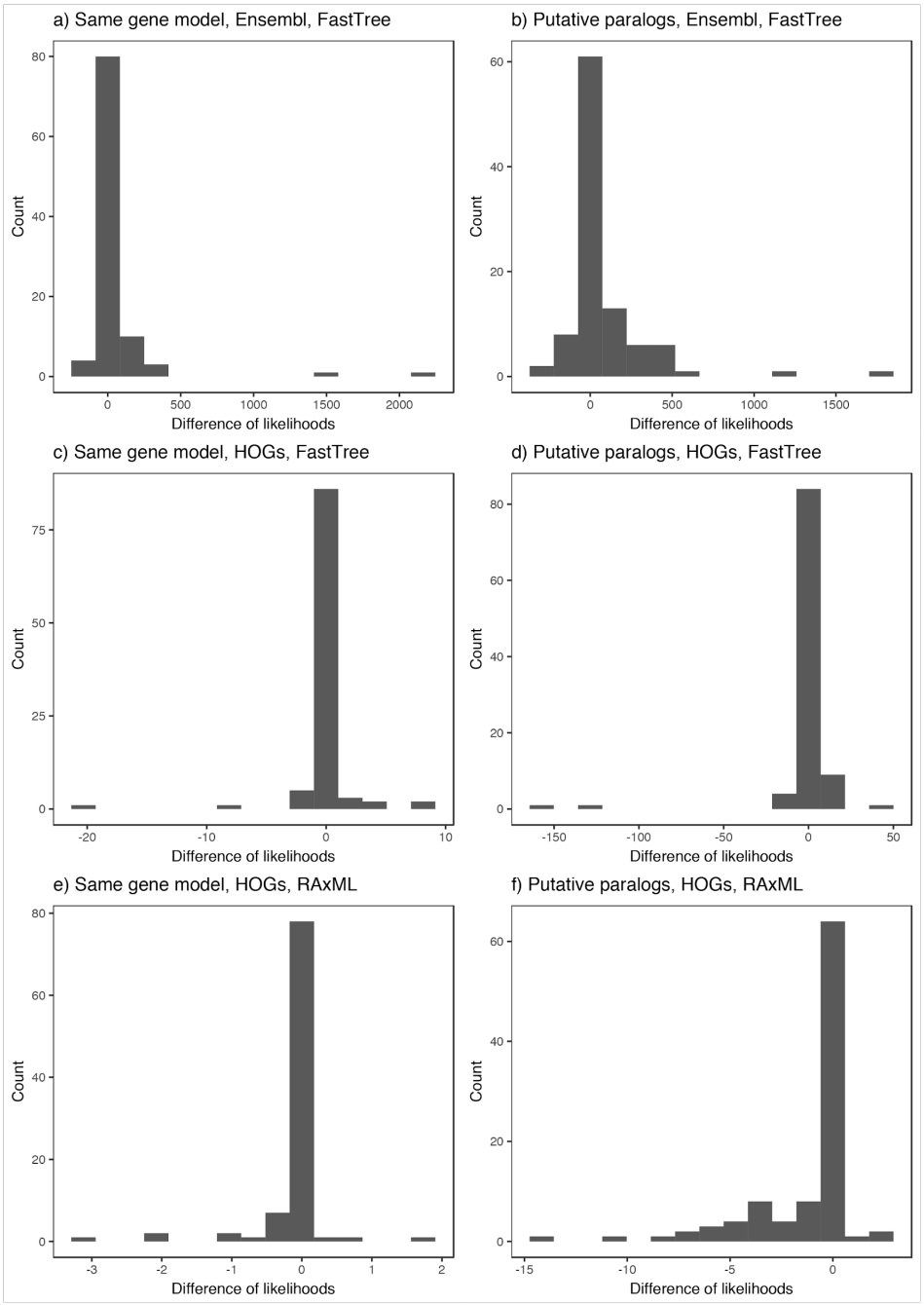


Figure B.9: Difference in likelihoods: likelihood of the ML tree starting from an input topology - likelihood of the ML tree without an input topology.

Cases depicted in the left column were derived from the same gene model while those depicted in the column on the right were derived from putative paralogous gene models.

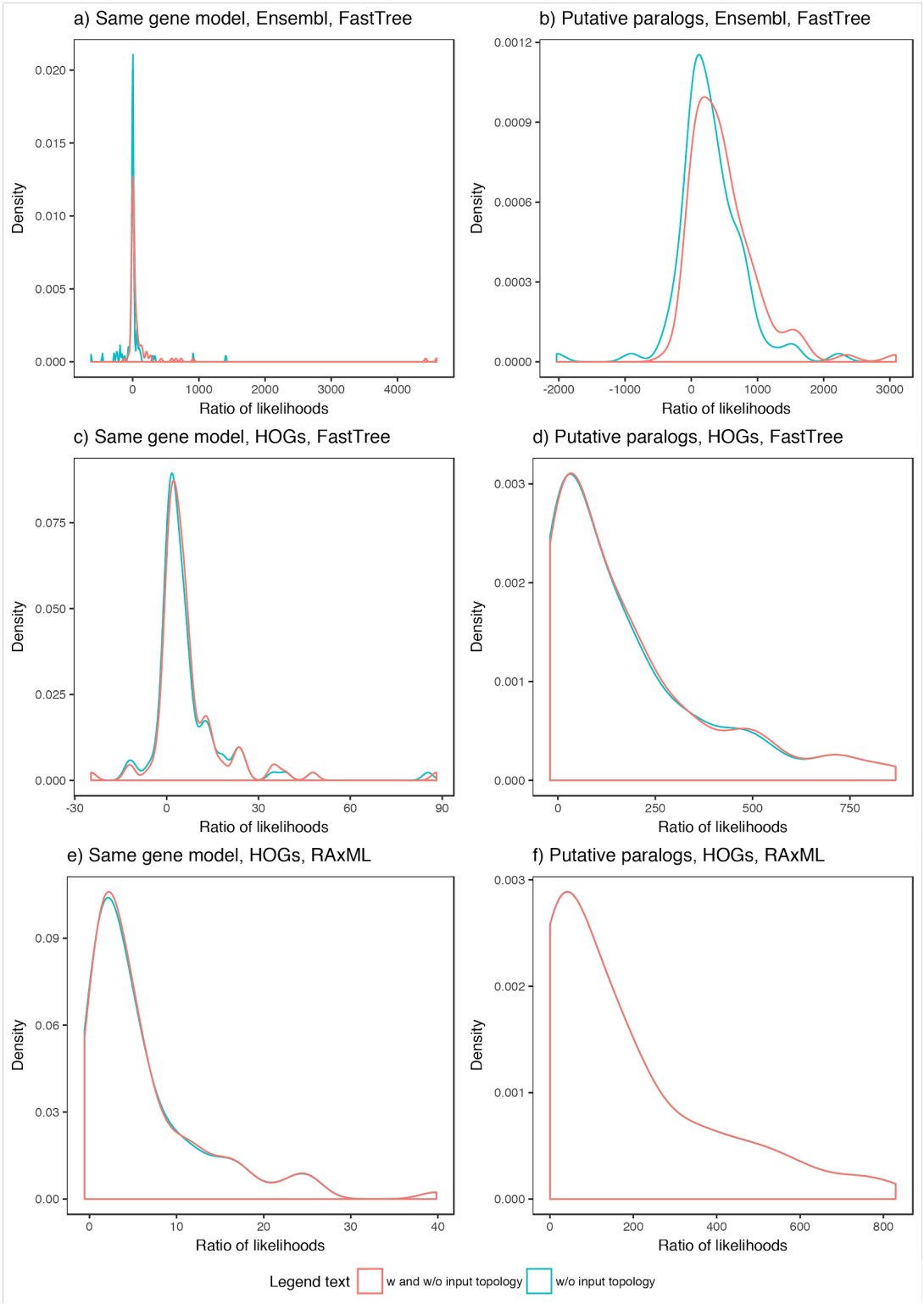


Figure B.10: Distributions of the ratio of likelihoods value.

Cases depicted in the left column were derived from the same gene model while those depicted in the column on the right were derived from putative paralogous gene models.

Appendix C

Detecting fragmented gene models in the putative genome of wild olive, with step-by-step assessments

C.1 Dataset

Table C.1: Putative proteomes exported from OMA Browser (Altenhoff *et al.*, 2018) and used as input data for GETHOGs algorithm (Altenhoff *et al.*, 2013) in the study on *Olea europaea* var. *sylvestris*.

The second column contains information on the database release that OMA Browser retrieved an assembly and annotation from.

Species	**Database**
Amborella trichopoda	Ensembl Plants 23
Arabidopsis thaliana	Ensembl Plants 38
Glycine max	Ensembl Plants 19
Helianthus annuus	Ensembl Plants 39
Oryza sativa subsp. japonica	Ensembl Plants 27
Populus trichocarpa	Ensembl Plants 15
Solanum lycopersicum	Ensembl Plants 27
Solanum tuberosum	Ensembl Plants 21
Vitis vinifera	Ensembl Plants v9
Zea mays	Ensembl Plants 40

C.2 Inspection of selected predictions

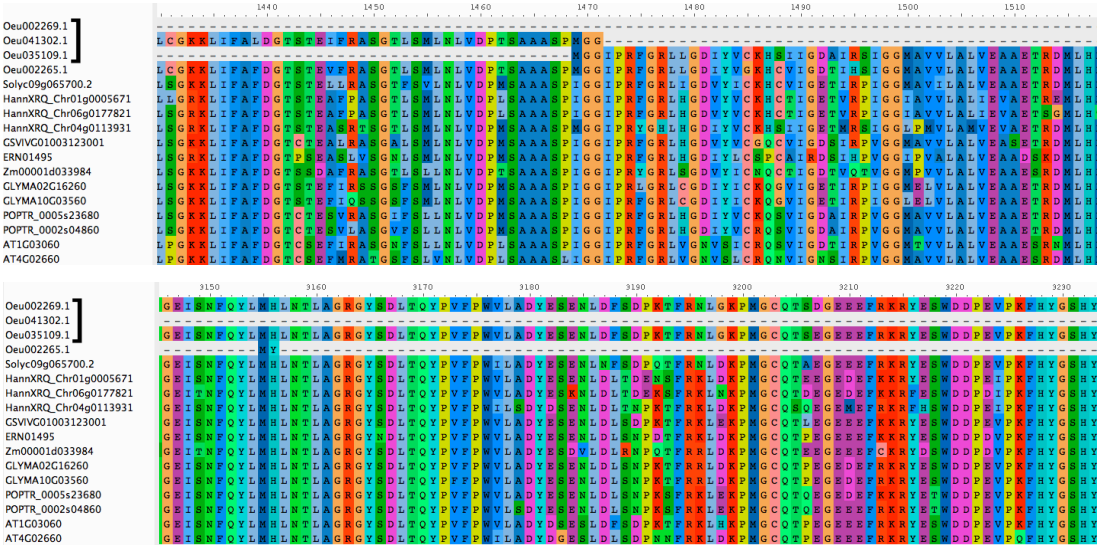


Figure C.1: Parts of 3,783 positions long multiple sequence alignment of a putative protein family yielding an ambiguous prediction for gene models coding putative protein sequences (Oeu002269.1, Oeu041302.1) (drawn with AliView (Larsson, 2014)).

Annotated protein sequences Oeu002269.1 and Oeu041302.1 are depicted in the first two rows while the third row depicts annotated sequence Oeu035109.1 involved in an ambiguous prediction (Oeu041302.1, Oeu035109.1).

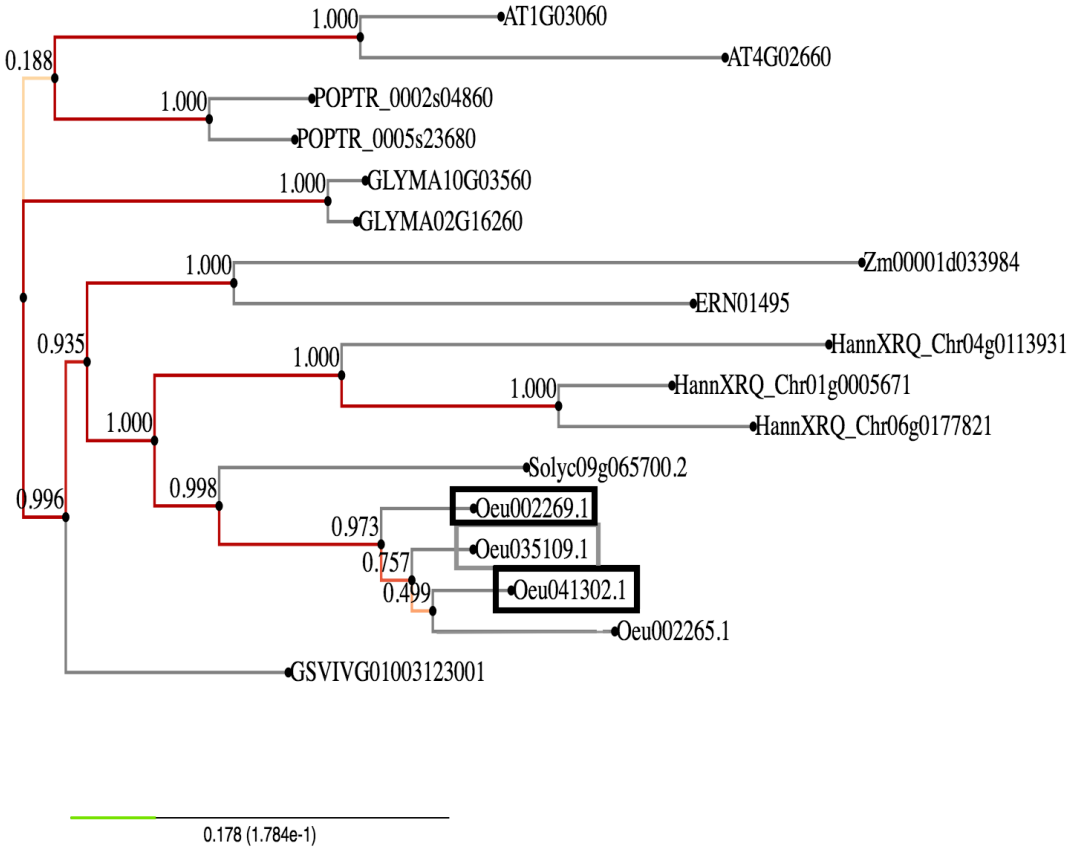


Figure C.2: Reconstructed protein tree with SH-like branch supports for multiple sequence alignment depicted in Figure C.1 (drawn with Phylo.io (Robinson, Dylus and Dessimoz, 2016)).

Black rectangles mark placement of putative protein sequences involved in an ambiguous prediction (Oeu002269.1, Oeu041302.1). Another ambiguous prediction from this putative family was (Oeu041302.1, Oeu035109.1).

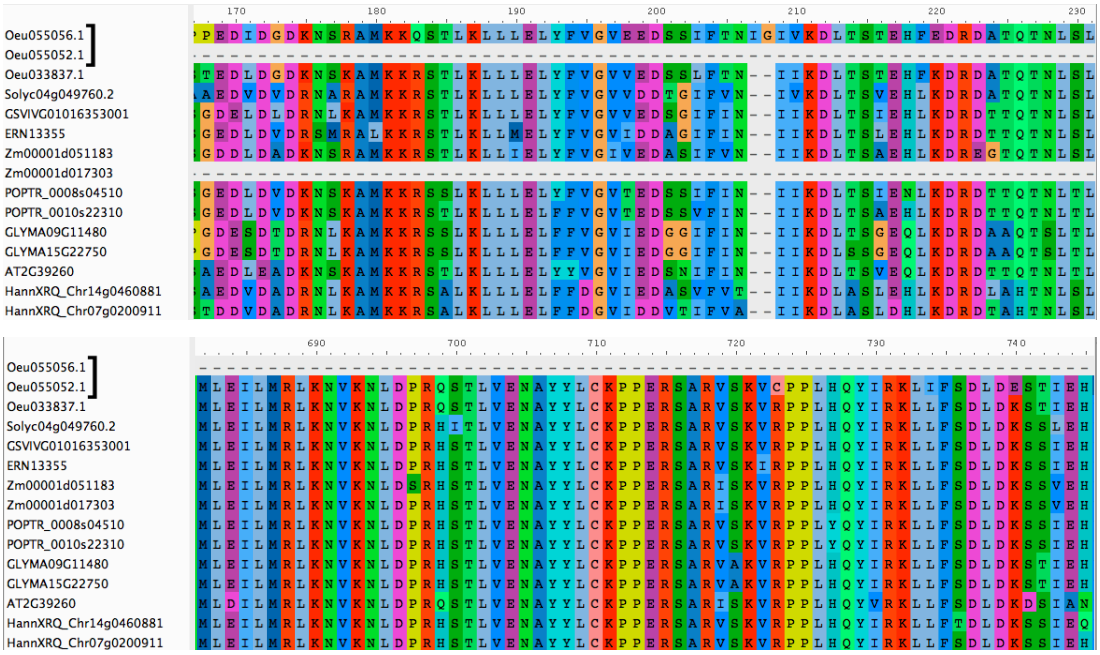


Figure C.3: Parts of 1,280 positions long multiple sequence alignment of a putative protein family yielding an unambiguous prediction for gene models coding putative protein sequences (Oeu055052.1, Oeu055056.1) (drawn with AliView (Larsson, 2014)).

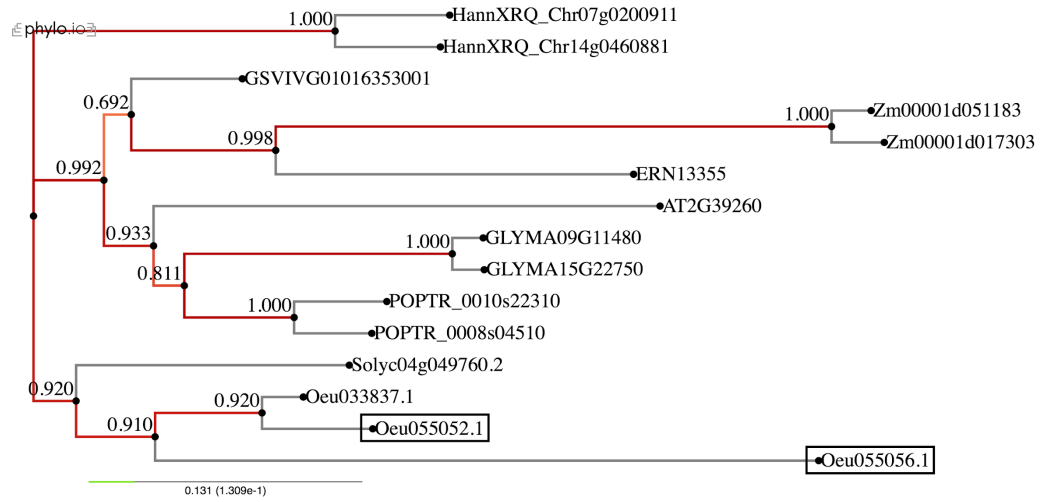


Figure C.4: Reconstructed protein tree with SH-like branch supports for multiple sequence alignment depicted in Figure C.3 (drawn with Phylo.io (Robinson, Dylus and Dessimoz, 2016)).

Black rectangles mark placement of putative protein sequences involved in an unambiguous prediction (Oeu055052.1, Oeu055056.1).

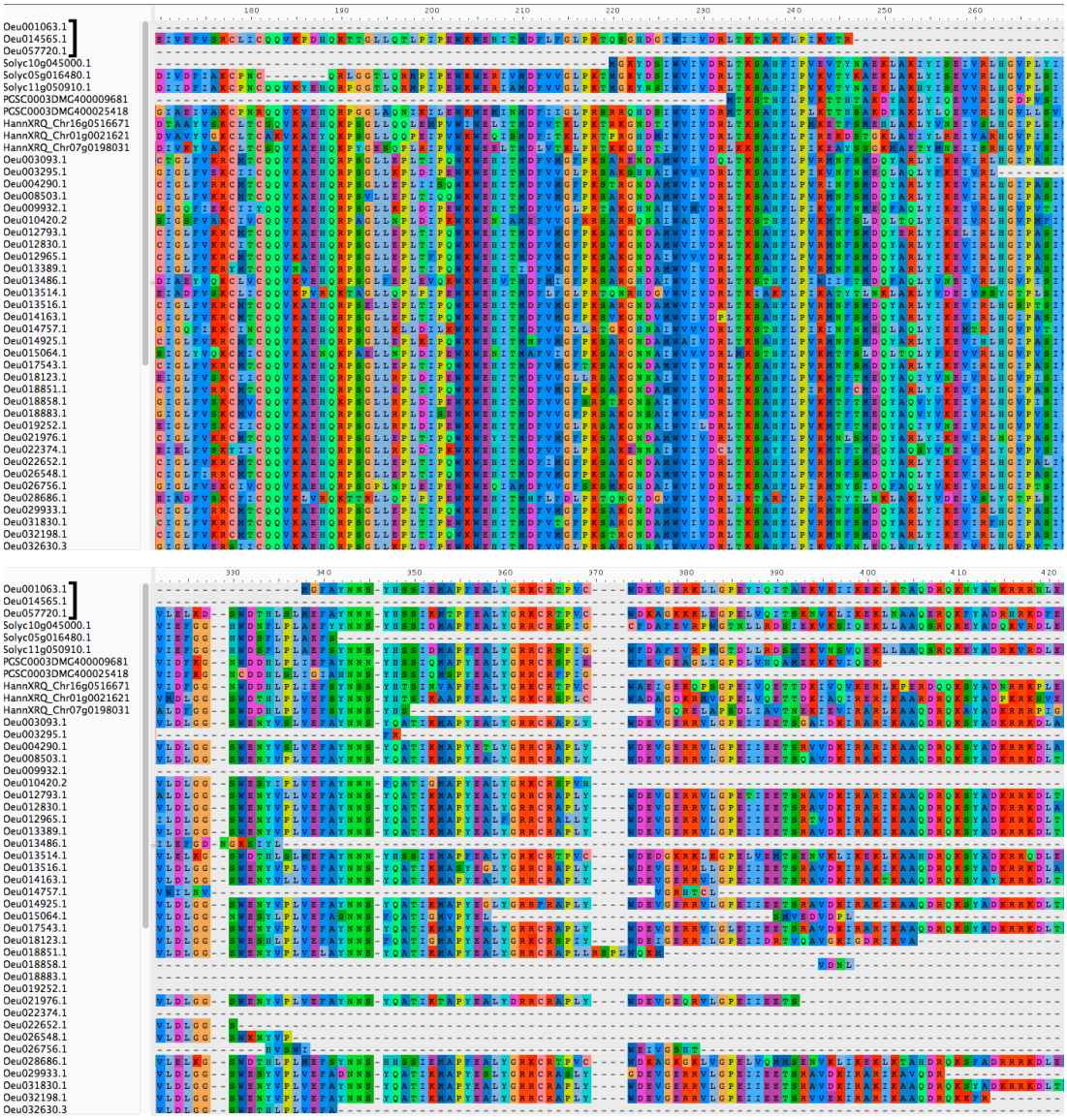


Figure C.5: Parts of 600 positions long multiple sequence alignment of a putative protein family yielding an ambiguous prediction for gene models coding putative protein sequences (Oeu001063.1, Oeu014565.1) (drawn with AliView (Larsson, 2014)).

Putative protein sequences Oeu001063.1 and Oeu014565.1 are depicted in the first two rows while the third row depicts annotated protein sequence Oeu057720.1 involved in an ambiguous prediction (Oeu057720.1, Oeu014565.1).

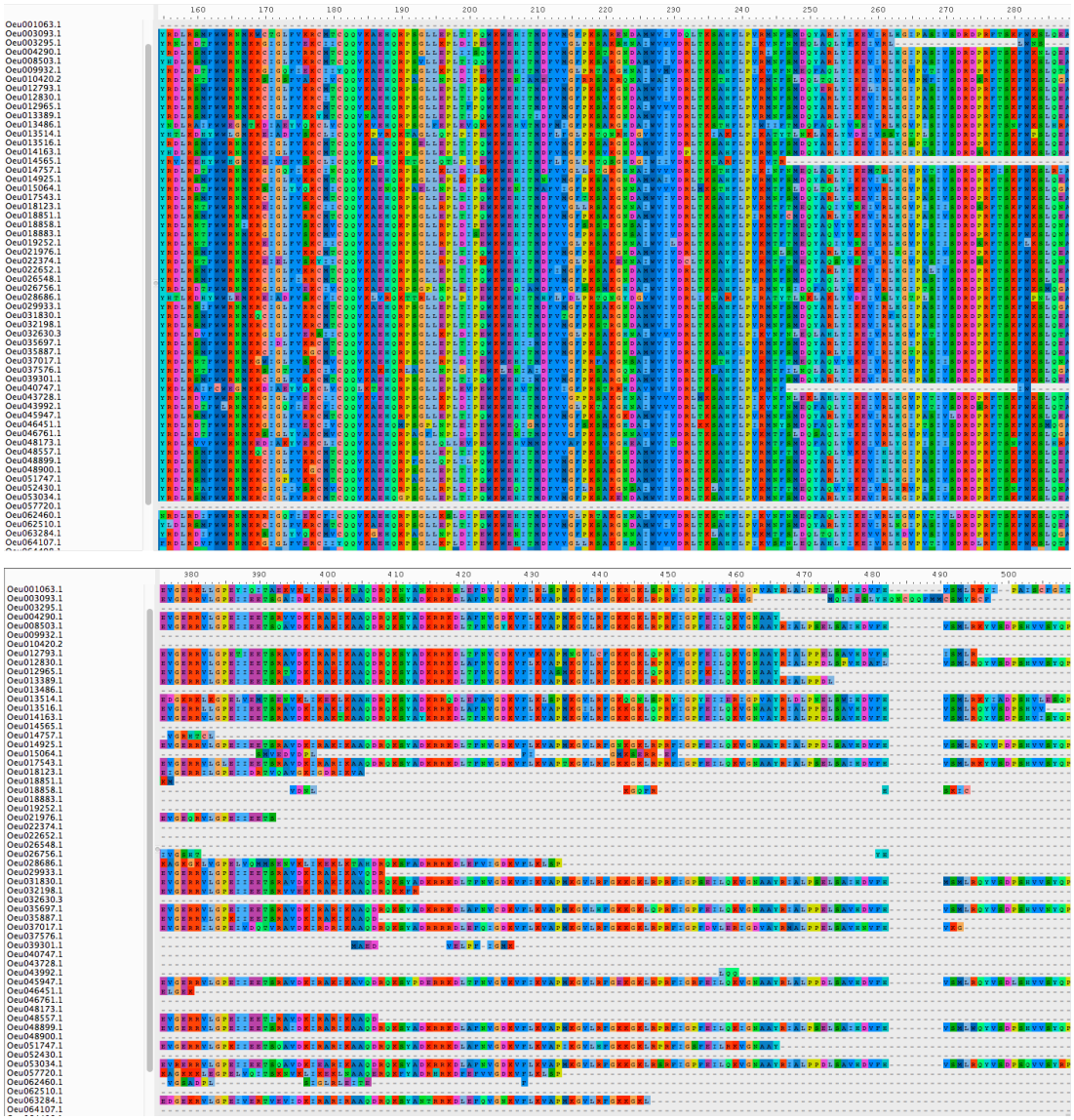


Figure C.6: Parts of 600 positions long multiple sequence alignment of a putative protein family providing an ambiguous prediction for gene models coding putative protein sequences (Oeu001063.1, Oeu014565.1)—wild olive putative sequences only (drawn with AliView (Larsson, 2014)).

Possible high duplication rate of the gene family coupled with potentially high heterozygosity rate and missing data.

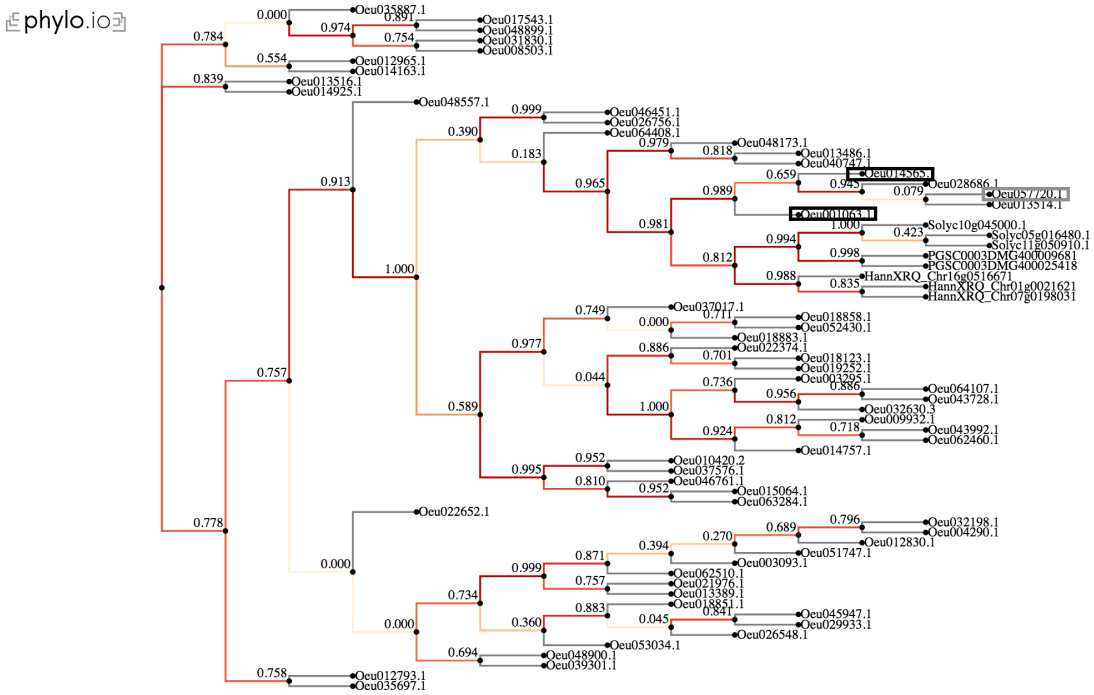


Figure C.7: Reconstructed protein tree with SH-like branch supports for multiple sequence alignment depicted in Figures C.5-C.6 (drawn with Phylo.io (Robinson, Dylus and Dessimoz, 2016)).

Black rectangles mark placement of putative proteins involved in an ambiguous prediction (Oeu001063.1, Oeu014565.1). Twenty four more cases from this putative family were examined and a pair of putative proteins (Oeu057720.1, Oeu014565.1) indicated another ambiguous split gene model.

Appendix D

Identifying fragments of the same transcript model in transcriptome datasets, with putative cassava transcriptome as a test case

D.1 Dataset

Table D.1: Putative proteomes exported from OMA Browser (Altenhoff *et al.*, 2014; Altenhoff *et al.*, 2018), March 2017 release and used as a set of references in the study.

They were five closest species to *Manihot esculenta* available in the database[55]. The second column contains information on the database release that OMA Browser retrieved an assembly and annotation from.

Species	**Database**
Arabidopsis thaliana	Ensembl Plants 20
Lotus japonicus	The Baylor College of Medicine—Human Genome Sequencing Center[56]
Medicago truncatula	Ensembl Plants 18
Populus trichocarpa	Ensembl Plants 15
Glycine max	Ensembl Plants 19

[55] As depicted in the reconstructed phylogenetic tree in OMA Browser

[56] The center also sequences species other than human. This putative proteome (Lotus japonicus v3.0) is no longer available for download from their website but can be downloaded from, e.g. *Lotus* Base (Mun *et al.*, 2016).

List of references

Abbe, E. (1863) *Über die Gesetzmäßigkeit in der Vertheilung der Fehler bei Beobachtungsrei*. Jena: Friedrich Fromman.

Abdel-Ghany, S. E. *et al.* (2016) 'A survey of the sorghum transcriptome using single-molecule long reads', *Nature Communications*, 7, p. 11706.

Acuña-Amador, L. *et al.* (2018) 'Genomic repeats, misassembly and reannotation: a case study with long-read resequencing of Porphyromonas gingivalis reference strains', *BMC Genomics*, 19(1), p. 54.

Adachi, J. and Hasegawa, M. (1996) *MOLPHY: Programs for molecular phylogenetics based on maximum likelihood, vers. 2.3*. Tokyo: Institute of Statistical Mathematics.

Adams, K. L. and Wendel, J. F. (2005) 'Polyploidy and genome evolution in plants', *Current Opinion in Plant Biology*, 8(2), pp. 135–141.

Ahlquist, P. *et al.* (1985) 'Sindbis virus proteins nsP1 and nsP2 contain homology to nonstructural proteins from several RNA plant viruses', *Journal of Virology*, 53(2), pp. 536–542.

Albalat, R. and Cañestro, C. (2016) 'Evolution by gene loss', *Nature Reviews Genetics*, 17(7), pp. 379–391.

Alexander, R. P. *et al.* (2010) 'Annotating non-coding regions of the genome', *Nature Reviews Genetics*, 11(8), pp. 559–571.

Alföldi, J. and Lindblad-Toh, K. (2013) 'Comparative genomics as a tool to understand evolution and disease', *Genome Research*, 23(7), pp. 1063–1068.

Alkan, C., Sajjadian, S. and Eichler, E. E. (2011) 'Limitations of next-generation genome sequence assembly', *Nature Methods*, 8(1), pp. 61–65.

Allen, J. E., Pertea, M. and Salzberg, S. L. (2004) 'Computational gene

prediction using multiple sources of evidence', *Genome Research*, 14(1), pp. 142–148.

Altenhoff, A. M. *et al.* (2010) 'OMA 2011: orthology inference among 1000 complete genomes', *Nucleic Acids Research*, 39(Database issue), pp. D289–D294.

Altenhoff, A. M. *et al.* (2012) 'Resolving the ortholog conjecture: orthologs tend to be weakly, but significantly, more similar in function than paralogs', *PLOS Computational Biology*, 8(5), p. e1002514.

Altenhoff, A. M. *et al.* (2013) 'Inferring hierarchical orthologous groups from orthologous gene pairs', *PLOS ONE*, 8(1), p. e53786.

Altenhoff, A. M. *et al.* (2014) 'The OMA orthology database in 2015: function predictions, better plant support, synteny view and other improvements', *Nucleic Acids Research*, 43(Database issue), pp. D240–D249.

Altenhoff, A. M. *et al.* (2018) 'The OMA orthology database in 2018: retrieving evolutionary relationships among all domains of life through richer web and programmatic interfaces', *Nucleic Acids Research*, 46(Database issue), pp. D477–D485.

Altenhoff, A. M. *et al.* (2019) 'OMA standalone: orthology inference among public and custom genomes and transcriptomes', *Genome Research*, 29(7), pp. 1152–1163.

Altenhoff, A. M. and Dessimoz, C. (2012) 'Inferring Orthology and Paralogy', in Anisimova, M. (ed.) *Evolutionary Genomics*. Totowa: Humana Press (Methods in Molecular Biology), pp. 259–279.

Altschul, S. F. *et al.* (1990) 'Basic local alignment search tool', *Journal of Molecular Biology*, 215(3), pp. 403–410.

Altschul, S. F. *et al.* (1997) 'Gapped BLAST and PSI-BLAST: a new generation of protein database search programs', *Nucleic Acids Research*, 25(17), pp. 3389–3402.

Andrews, S. (2010) *FastQC: a quality control tool for high throughput sequence data*. Babraham Bioinformatics, Babraham Institute, Cambridge, United Kingdom. Available at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc.

ARC-OVI (2015) *BioProject PRJNA282938*, *NCBI*. Available at: https://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRX2013075 (Accessed: 12 August 2017).

Argos, P. *et al.* (1984) 'Similarity in gene organization and homology between proteins of animal picomaviruses and a plant comovirus suggest common ancestry of these virus families', *Nucleic Acids Research*, 12(18), pp. 7251–7267.

Arkin Lab (2008) *FastTree*, *MicrobesOnline*. Available at: http://www.microbesonline.org/fasttree (Accessed: 24 May 2019).

Arkin Lab (2008) *FastTree Change Log*, *MicrobesOnline*. Available at: http://www.microbesonline.org/fasttree/ChangeLog (Accessed: 14 June 2019).

Armstrong, J. *et al.* (2019) 'Whole-Genome Alignment and Comparative Annotation', *Annual Review of Animal Biosciences*, 7, pp. 41–64.

Awoleye, F. *et al.* (1994) 'Nuclear DNA content and in vitro induced somatic polyploidization cassava (Manihot esculenta Crantz) breeding', *Euphytica*, 76(3), pp. 195–202.

Babarinde, I. A., Li, Y. and Hutchins, A. P. (2019) 'Computational Methods for Mapping, Assembly and Quantification for Coding and Non-coding Transcripts', *Computational and Structural Biotechnology Journal*, 17, pp. 628–637.

Babraham Institute Bioinformatics Group (no date) *FastQC Help Pages*, *Babraham Bioinformatics*. Available at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules (Accessed: 10 July 2019).

Baker, M. (2012) ‘De novo genome assembly: what every biologist should know’, *Nature Methods*, 9(4), pp. 333–337.

Baldauf, S. L. (2003) ‘Phylogeny for the faint of heart: a tutorial’, *Trends in Genetics*, 19(6), pp. 345–351.

Barghini, E. *et al.* (2014) ‘The peculiar landscape of repetitive sequences in the olive (Olea europaea L.) genome’, *Genome Biology and Evolution*, 6(4), pp. 776–791.

Bart, R. S. and Taylor, N. J. (2017) ‘New opportunities and challenges to engineer disease resistance in cassava, a staple food of African small-holder farmers’, *PLOS Pathogens*, 13(5), p. e1006287.

Benjamini, Y. and Speed, T. P. (2012) ‘Summarizing and correcting the GC content bias in high-throughput sequencing’, *Nucleic Acids Research*, 40(10), p. e72.

Benson, D. A. *et al.* (2012) ‘GenBank’, *Nucleic Acids Research*, 41(Database issue), pp. D36–D42.

Beretta, S. *et al.* (2018) ‘HapCHAT: adaptive haplotype assembly for efficiently leveraging high coverage in long reads’, *BMC Bioinformatics*, 19(1), p. 252.

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

Bernardes, J. S., Carbone, A. and Zaverucha, G. (2011) ‘A discriminative method for family-based protein remote homology detection that combines inductive logic programming and propositional models’, *BMC Bioinformatics*, 12, p. 83.

Berretta, J. and Morillon, A. (2009) ‘Pervasive transcription constitutes a new level of eukaryotic genome regulation’, *EMBO Reports*, 10(9), pp. 973–982.

Besnard, G., Terral, J.-F. and Cornille, A. (2017) ‘On the origins and

domestication of the olive: a review and perspectives', *Annals of Botany*, 121(3), pp. 385–403.

Besser, J. *et al.* (2018) 'Next-generation sequencing technologies and their application to the study and control of bacterial infections', *Clinical Microbiology and Infection*, 24(4), pp. 335–341.

Bevan, M. W. *et al.* (2017) 'Genomic innovation for crop improvement', *Nature*, 543(7645), pp. 346–354.

Bhadauria, V. (2017) *Next-generation Sequencing and Bioinformatics for Plant Science*. Norfolk: Caister Academic Press.

Biscotti, M. A., Olmo, E. and Heslop-Harrison, J. S. P. (2015) 'Repetitive DNA in eukaryotic genomes', *Chromosome Research*, 23(3), pp. 415–420.

Boeckmann, B. *et al.* (2011) 'Conceptual framework and pilot study to benchmark phylogenomic databases based on reference gene trees', *Briefings in Bioinformatics*, 12(5), pp. 423–435.

Boetzer, M. *et al.* (2011) 'Scaffolding pre-assembled contigs using SSPACE', *Bioinformatics*, 27(4), pp. 578–579.

Boetzer, M. and Pirovano, W. (2012) 'Toward almost closed genomes with GapFiller', *Genome Biology*, 13(6), p. R56.

Bokulich, N. A. *et al.* (2013) 'Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing', *Nature Methods*, 10(1), pp. 57–59.

Bolger, M. *et al.* (2017) 'From plant genomes to phenotypes', *Journal of Biotechnology*, 261, pp. 46–52.

Bolger, M. E., Arsova, B. and Usadel, B. (2018) 'Plant genome and transcriptome annotations: from misconceptions to simple solutions', *Briefings in Bioinformatics*, 19(3), pp. 437–449.

Bolisetty, M. T., Rajadinakaran, G. and Graveley, B. R. (2015) 'Determining

exon connectivity in complex mRNAs by nanopore sequencing', *Genome Biology*, 16, p. 204.

Bongartz, P. (2019) 'Resolving repeat families with long reads', *BMC Bioinformatics*, 20(1), p. 232.

Bornberg-Bauer, E. *et al.* (2005) 'The evolution of domain arrangements in proteins and interaction networks', *Cellular and Molecular Life Sciences*, 62(4), pp. 435–445.

Bradnam, K. R. *et al.* (2013) 'Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species', *GigaScience*, 2(1).

Bräutigam, A. *et al.* (2011) 'An mRNA blueprint for C4 photosynthesis derived from comparative transcriptomics of closely related C3 and C4 species', *Plant Physiology*, 155(1), pp. 142–156.

Bredeson, J. V. *et al.* (2016) 'Sequencing wild and cultivated cassava and related species reveals extensive interspecific hybridization and genetic diversity', *Nature Biotechnology*, 34(5), pp. 562–570.

Brenchley, R. *et al.* (2012) 'Analysis of the bread wheat genome using whole-genome shotgun sequencing', *Nature*, 491(7426), pp. 705–710.

Brendel, V. *et al.* (1992) 'Methods and algorithms for statistical analysis of protein sequences', *Proceedings of the National Academy of Sciences of the United States of America*, 89(6), pp. 2002–2006.

Brent, M. R. (2005) 'Genome annotation past, present, and future: How to define an ORF at each locus', *Genome Research*, 15(12), pp. 1777–1786.

Burton, J. N. *et al.* (2013) 'Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions', *Nature Biotechnology*, 31(12), pp. 1119–1125.

Bushmanova, E. *et al.* (2016) 'rnaQUAST: a quality assessment tool for de novo transcriptome assemblies', *Bioinformatics*, 32(14), pp. 2210–2212.

Butler, J. *et al.* (2008) ‘ALLPATHS: de novo assembly of whole-genome shotgun microreads’, *Genome Research*, 18(5), pp. 810–820.

Butlin, R. K., Galindo, J. and Grahame, J. W. (2008) ‘Sympatric, parapatric or allopatric: the most important way to classify speciation?’, *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 363(1506), pp. 2997–3007.

Byrne, A. *et al.* (2019) ‘Realizing the potential of full-length transcriptome sequencing’, *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 374(1786), p. 20190097.

Byrne, K. P. and Wolfe, K. H. (2005) ‘The Yeast Gene Order Browser: combining curated homology and syntenic context reveals gene fate in polyploid species’, *Genome Research*, 15(10), pp. 1456–1461.

Camacho, C. *et al.* (2009) ‘BLAST+: architecture and applications’, *BMC Bioinformatics*, 10, p. 421.

Cao, H. *et al.* (2015) ‘De novo assembly of a haplotype-resolved human genome’, *Nature Biotechnology*, 33(6), pp. 617–622.

Carmona, R. *et al.* (2015) ‘ReprOlive: a database with linked data for the olive tree (Olea europaea L.) reproductive transcriptome’, *Frontiers in Plant Science*, 6, p. 625.

Cavalli-Sforza, L. L. and Edwards, A. W. F. (1967) ‘Phylogenetic analysis: models and estimation procedures’, *Evolution*, 21(3), pp. 550–570.

Chaisson, M. J. P., Wilson, R. K. and Eichler, E. E. (2015) ‘Genetic variation and the de novo assembly of human genomes’, *Nature Reviews Genetics*, 16(11), pp. 627–640.

Chapman, J. A. *et al.* (2015) ‘A whole-genome shotgun approach for assembling and anchoring the hexaploid bread wheat genome’, *Genome Biology*, 16, p. 26.

Chatterji, S. and Pachter, L. (2006) ‘Reference based annotation with

GeneMapper', *Genome Biology*, 7(4), p. R29.

Chekanova, J. A. and Wang, H.-L. V. (2019) *Plant Long Non-Coding RNAs: Methods and Protocols*. New York: Humana Press.

Cheng, C.-Y. *et al.* (2017) 'Araport11: a complete reannotation of the Arabidopsis thaliana reference genome', *The Plant Journal*, 89(4), pp. 789–804.

Chen, G., Shi, T. and Shi, L. (2017) 'Characterizing and annotating the genome using RNA-seq data', *Science China Life Sciences*, 60(2), pp. 116–125.

Chen, J. *et al.* (2016) 'A comprehensive review and comparison of different computational methods for protein remote homology detection', *Briefings in Bioinformatics*, 19(2), pp. 231–244.

Chen, J., Liu, B. and Huang, D. (2016) 'Protein Remote Homology Detection Based on an Ensemble Learning Approach', *BioMed Research International*, 2016, p. 5813645.

Chen, K. and Kurgan, L. (2007) 'PFRES: protein fold classification by using evolutionary information and predicted secondary structure', *Bioinformatics*, 23(21), pp. 2843–2850.

Chen, S.-M., Ma, K.-Y. and Zeng, J. (2011) 'Pseudogene: lessons from PCR bias, identification and resurrection', *Molecular Biology Reports*, 38(6), pp. 3709–3715.

Chen, Y.-C. *et al.* (2013) 'Effects of GC bias in next-generation-sequencing data on de novo genome assembly', *PLOS ONE*, 8(4), p. e62856.

Childs, K. L. (2014) 'Methods for Plant Genome Annotation', in Bell, E. (ed.) *Molecular Life Sciences*. New York: Springer.

Chin, C.-S. *et al.* (2016) 'Phased diploid genome assembly with single-molecule real-time sequencing', *Nature Methods*, 13(12), pp. 1050–1054.

Choi, S. and Wing, R. A. (2000) ‘The Construction of Bacterial Artificial Chromosome (BAC) Libraries’, in Gelvin, S. B. and Schilperoort, R. A. (eds) *Plant Molecular Biology Manual*. Dordrecht: Springer, pp. 1–28.

Cho, N.-H. *et al.* (2007) ‘The Orientia tsutsugamushi genome reveals massive proliferation of conjugative type IV secretion system and host-cell interaction genes’, *Proceedings of the National Academy of Sciences of the United States of America*, 104(19), pp. 7981–7986.

Choo, K. H., Tong, J. C. and Zhang, L. (2004) ‘Recent applications of Hidden Markov Models in computational biology’, *Genomics, Proteomics & Bioinformatics*, 2(2), pp. 84–96.

Choulet, F. *et al.* (2014) ‘Structural and functional partitioning of bread wheat chromosome 3B’, *Science*, 345(6194), p. 1249721.

Claros, M. G. *et al.* (2012) ‘Why assembling plant genome sequences is so challenging’, *Biology*, 1(2), pp. 439–459.

Clavijo, B. J. *et al.* (2017) ‘An improved assembly and annotation of the allohexaploid wheat genome identifies complete families of agronomic genes and provides genomic evidence for chromosomal translocations’, *Genome Research*, 27(5), pp. 885–896.

Collingridge, P. W. and Kelly, S. (2012) ‘MergeAlign: improving multiple sequence alignment performance by dynamic reconstruction of consensus multiple sequence alignments’, *BMC Bioinformatics*, 13, p. 117.

Conesa, A. *et al.* (2016) ‘A survey of best practices for RNA-seq data analysis’, *Genome Biology*, 17, p. 13.

Cook, D. E. *et al.* (2019) ‘Long-Read Annotation: Automated Eukaryotic Genome Annotation Based on Long-Read cDNA Sequencing’, *Plant Physiology*, 179(1), pp. 38–54.

Costa, V. *et al.* (2010) ‘Uncovering the complexity of transcriptomes with RNA-Seq’, *Journal of Biomedicine & Biotechnology*, 2010, p. 853916.

Coyne, J. A. and Allen Orr, H. (2004) *Speciation*. Sunderland: Sinauer Associates.

Cruz, F. *et al.* (2016) 'Genome sequence of the olive tree, Olea europaea', *GigaScience*, 5(1), p. 29.

Cunningham, F. *et al.* (2019) 'Ensembl 2019', *Nucleic Acids Research*, 47(Database issue), pp. D745–D751.

Dalquen, D. A. *et al.* (2012) 'ALF—A Simulation Framework for Genome Evolution', *Molecular Biology and Evolution*, 29(4), pp. 1115–1123.

Dalquen, D. A. *et al.* (2013) 'The impact of gene duplication, insertion, deletion, lateral gene transfer and sequencing error on orthology inference: a simulation study', *PLOS ONE*, 8(2), p. e56925.

Darwin, C. (1859) *On the origins of species by means of natural selection*. London: John Murray.

Dayhoff, M. O., Schwartz, R. M. and Orcutt, B. C. (1978) 'A Model of Evolutionary Change in Proteins', in Dayhoff, M. O. (ed.) *Atlas of Protein Sequence and Structure*. Washington, DC: National Biomedical Research Foundation (5), pp. 345–352.

Dehal, P. S. *et al.* (2010) 'MicrobesOnline: an integrated portal for comparative and functional genomics', *Nucleic Acids Research*, 38(Database issue), pp. D396–D400.

Delignette-Muller, M. L., Dutang, C. and Others (2015) 'fitdistrplus: An R package for fitting distributions', *Journal of Statistical Software*, 64(4), pp. 1–34.

Delsuc, F., Brinkmann, H. and Philippe, H. (2005) 'Phylogenomics and the reconstruction of the tree of life', *Nature Reviews Genetics*, 6(5), p. 361.

De Maio, N. *et al.* (2019) 'Comparison of long-read sequencing technologies in the hybrid assembly of complex bacterial genomes', *Microbial Genomics*, 5(9), p. e000294.

Denton, J. F. *et al.* (2014) ‘Extensive error in the number of genes inferred from draft genome assemblies’, *PLOS Computational Biology*, 10(12), p. e1003998.

Desper, R. and Gascuel, O. (2002) ‘Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle’, *Journal of Computational Biology*, 9(5), pp. 687–705.

Dessimoz, C. *et al.* (2005) ‘OMA, A Comprehensive, Automated Project for the Identification of Orthologs from Complete Genome Data: Introduction and First Achievements’, in McLysaght, A. and Huson, D. H. (eds) *Comparative Genomics. RCG 2005. Lecture Notes in Computer Science*. Heidelberg: Springer, Berlin, Heidelberg, pp. 61–72.

Dessimoz, C. *et al.* (2011) ‘Comparative genomics approach to detecting split-coding regions in a low-coverage genome: lessons from the chimaera Callorhinchus milii (Holocephali, Chondrichthyes)’, *Briefings in Bioinformatics*, 12(5), pp. 474–484.

van Dijk, E. L. *et al.* (2018) ‘The Third Revolution in Sequencing Technology’, *Trends in Genetics*, 34(9), pp. 666–681.

Djebali, S. *et al.* (2012) ‘Landscape of transcription in human cells’, *Nature*, 489(7414), pp. 101–108.

Dlugosch, K. M. *et al.* (2013) ‘Allele identification for transcriptome-based population genomics in the invasive plant Centaurea solstitialis’, *G3*, 3(2), pp. 359–367.

Doležel, J. *et al.* (2012) ‘Chromosomes in the flow to simplify genome analysis’, *Functional & Integrative Genomics*, 12(3), pp. 397–416.

Dominguez Del Angel, V. *et al.* (2018) ‘Ten steps to get started in Genome Assembly and Annotation [version 1; peer review: 2 approved]’, *F1000Research*, 7(ELIXIR), p. 148.

Dong, Y. *et al.* (2013) ‘Sequencing and automated whole-genome optical

mapping of the genome of a domestic goat (Capra hircus)', *Nature Biotechnology*, 31(2), pp. 135–141.

Doyon, J.-P. *et al.* (2011) 'Models, algorithms and programs for phylogeny reconciliation', *Briefings in Bioinformatics*, 12(5), pp. 392–400.

Drummond, A. J. *et al.* (2006) 'Relaxed phylogenetics and dating with confidence', *PLOS Biology*, 4(5), p. e88.

Dunn, C. W. and Munro, C. (2016) 'Comparative genomics and the diversity of life', *Zoologica Scripta*, 45(S1), pp. 5–13.

Durbin, R. *et al.* (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge: Cambridge University Press.

Earl, D. *et al.* (2011) 'Assemblathon 1: a competitive assessment of de novo short read assembly methods', *Genome Research*, 21(12), pp. 2224–2241.

Eckersley-Maslin, M. A. and Spector, D. L. (2014) 'Random monoallelic expression: regulating gene expression one allele at a time', *Trends in Genetics*, 30(6), pp. 237–244.

Eddy, S. R. (1996) 'Hidden Markov models', *Current Opinion in Structural Biology*, 6(3), pp. 361–365.

Eddy, S. R. (1998) 'Profile hidden Markov models', *Bioinformatics*, 14(9), pp. 755–763.

Eddy, S. R. (2011) 'Accelerated Profile HMM Searches', *PLOS Computational Biology*, 7(10), p. e1002195.

Eddy, S. R. and the HMMER development team (2019) *HMMER User's Guide*. Howard Hughes Medical Institute. Available at: http://eddylab.org/software/hmmer/Userguide.pdf.

Edgar, R. C. (2004a) 'MUSCLE: a multiple sequence alignment method with reduced time and space complexity', *BMC Bioinformatics*, 5(1), p. 113.

Edgar, R. C. (2004b) 'MUSCLE: multiple sequence alignment with high

accuracy and high throughput', *Nucleic Acids Research*, 32(5), pp. 1792–1797.

Edgar, R. C. (2010) 'Search and clustering orders of magnitude faster than BLAST', *Bioinformatics*, 26(19), pp. 2460–2461.

Edgar, R. C. (no date) *UCLUST algorithm*. Available at: https://drive5.com/usearch/manual/uclust_algo.html.

Edgar, R. C. and Flyvbjerg, H. (2015) 'Error filtering, pair assembly and error correction for next-generation sequencing reads', *Bioinformatics*, 31(21), pp. 3476–3482.

Edgar, R. C. and Sjölander, K. (2004) 'A comparison of scoring functions for protein sequence profile alignment', *Bioinformatics*, 20(8), pp. 1301–1308.

Edge, P., Bafna, V. and Bansal, V. (2017) 'HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies', *Genome Research*, 27(5), pp. 801–812.

Edwards, A. W. F. (1972) *Likelihood*. Cambridge: Cambridge University Press.

Efron, B. and Tibshirani, R. J. (1993) 'Hypothesis testing with the bootstrap', in Efron, B. and Tibshirani, R. J. (eds) *An Introduction to the Bootstrap*. Boston: Springer US, pp. 220–236.

Eilbeck, K. *et al.* (2009) 'Quantitative measures for the management and comparison of annotated genomes', *BMC Bioinformatics*, 10, p. 67.

Ellegren, H. (2008) 'Comparative genomics and the study of evolution by natural selection', *Molecular Ecology*, 17(21), pp. 4586–4596.

EMBL-EBI (2017) *InterPro Consortium.* Available at: https://www.ebi.ac.uk/interpro/about/consortium (Accessed: 13 January 2020).

EMBL-EBI (2019a) *Protein trees.* Available at:

https://www.ensembl.org/info/genome/compara/homology_method.html (Accessed: 13 January 2020).

EMBL-EBI (2019b) *Protein trees and orthologies.* Available at: http://plants.ensembl.org/info/genome/compara/homology_method.html (Accessed: 1 November 2019).

EMBL-EBI (2019c) *TreeFam HMM library.* Available at: https://www.ensembl.org/info/genome/compara/hmm_lib.html (Accessed: 13 January 2020).

EMBL-EBI (2020) *Assembly: GCA_000001735.1*, *The European Nucleotide Archive (ENA)*. Available at: https://www.ebi.ac.uk/ena/data/view/GCA_000001735.1 (Accessed: 20 July 2019).

Estruch, R. *et al.* (2013) 'Primary prevention of cardiovascular disease with a Mediterranean diet', *The New England Journal of Medicine*, 368(14), pp. 1279–1290.

Estruch, R. *et al.* (2018) 'Primary Prevention of Cardiovascular Disease with a Mediterranean Diet Supplemented with Extra-Virgin Olive Oil or Nuts', *The New England Journal of Medicine*, 378(25), p. e34.

Eugster, P. T. *et al.* (2003) 'The Many Faces of Publish/Subscribe', *ACM Computing Surveys (CSUR)*, 35(2), pp. 114–131.

Falkhausen, M., Reininger, H. and Wolf, D. (1995) 'Calculation of distance measures between hidden Markov models', in *EUROSPEECH-1995*, pp. 1487–1490.

Fariselli, P. *et al.* (2007) 'The WWWH of remote homolog detection: the state of the art', *Briefings in Bioinformatics*, 8(2), pp. 78–87.

Farrell, J. D. *et al.* (2014) 'De novo assembly of the perennial ryegrass transcriptome using an RNA-Seq strategy', *PLOS ONE*, 9(8), p. e103567.

Felsenstein, J. (2004) *Inferring phylogenies*. Sunderland: Sinauer

Associates.

Felsenstein, J. (2005) *PHYLIP: Phylogenetic inference program*. Available at: http://evolution.genetics.washington.edu/phylip.html.

Feng, Y. *et al.* (2015) 'Nanopore-based fourth-generation DNA sequencing technology', *Genomics, Proteomics & Bioinformatics*, 13(1), pp. 4–16.

Filipczyk, A. *et al.* (2013) 'Biallelic expression of nanog protein in mouse embryonic stem cells', *Cell Stem Cell*, 13(1), pp. 12–13.

Finn, R. D. *et al.* (2014) 'Pfam: the protein families database', *Nucleic Acids Research*, 42(Database issue), pp. D222–D230.

Fitch, W. M. (1970) 'Distinguishing homologous from analogous proteins', *Systematic Zoology*, 19(2), pp. 99–113.

Fitch, W. M. (2000) 'Homology a personal view on some of the problems', *Trends in Genetics*, 16(5), pp. 227–231.

Fitzpatrick, B. M., Fordyce, J. A. and Gavrilets, S. (2009) 'Pattern, process and geographic modes of speciation', *Journal of Evolutionary Biology*, 22(11), pp. 2342–2347.

Freedman, A. (2016) *Best Practices for De Novo Transcriptome Assembly with Trinity*, *Harvard FAS Informatics*. Available at: http://informatics.fas.harvard.edu/best-practices-for-de-novo-transcriptome-assembly-with-trinity.html (Accessed: 12 September 2017).

Fryslie, B. (no date) *Newbler*. Available at: https://swes.cals.arizona.edu/maier_lab/kartchner/documentation/index.php/home/docs/newbler (Accessed: 26 June 2018).

Gabaldón, T. and Koonin, E. V. (2013) 'Functional and evolutionary implications of gene orthology', *Nature Reviews Genetics*, 14(5), pp. 360–366.

Garber, M. *et al.* (2011) 'Computational methods for transcriptome

annotation and quantification using RNA-seq', *Nature Methods*, 8(6), pp. 469–477.

Gaudet, P. *et al.* (2011) 'Phylogenetic-based propagation of functional annotations within the Gene Ontology consortium', *Briefings in Bioinformatics*, 12(5), pp. 449–462.

Gelman, A. *et al.* (2003) *Bayesian Data Analysis, Second Edition*. Boca Raton: CRC Press.

Geniza, M. and Jaiswal, P. (2017) 'Tools for building de novo transcriptome assembly', *Current Plant Biology*, 11(12), pp. 41–45.

Gerstein, M. B. *et al.* (2007) 'What is a gene, post-ENCODE? History and updated definition', *Genome Research*, 17(6), pp. 669–681.

Ghurye, J. *et al.* (2017) 'Scaffolding of long read assemblies using long range contact information', *BMC Genomics*, 18(1), p. 527.

Ghurye, J. and Pop, M. (2019) 'Modern technologies and algorithms for scaffolding assembled genomes', *PLOS Computational Biology*, 15(6), p. e1006994.

Gimelbrant, A. *et al.* (2007) 'Widespread monoallelic expression on human autosomes', *Science*, 318(5853), pp. 1136–1140.

Gish, W. R. (1996-2003) *WU BLAST*. Available at: http://blast.wustl.edu.

Glöckner, F. O. *et al.* (2017) '25 years of serving the community with ribosomal RNA gene reference databases and tools', *Journal of Biotechnology*, 261, pp. 169–176.

Glover, N. M., Redestig, H. and Dessimoz, C. (2016) 'Homoeologs: What Are They and How Do We Infer Them?', *Trends in Plant Science*, 21(7), pp. 609–621.

Gnerre, S. *et al.* (2011) 'High-quality draft assemblies of mammalian genomes from massively parallel sequence data', *Proceedings of the*

National Academy of Sciences of the United States of America, 108(4), pp. 1513–1518.

Gogarten, M. B., Gogarten, J. P. and Olendzenski, L. C. (eds) (2009) *Horizontal Gene Transfer: Genomes in Flux*. Humana Press.

Goldman, N. (1993) 'Statistical tests of models of DNA substitution', *Journal of Molecular Evolution*, 36(2), pp. 182–198.

Goloboff, P. A., Farris, J. S. and Nixon, K. C. (2008) 'TNT, a free program for phylogenetic analysis', *Cladistics*, 24(5), pp. 774–786.

Goltsman, E., Ho, I. and Rokhsar, D. (2017) 'Meraculous-2D: Haplotype-sensitive Assembly of Highly Heterozygous genomes', *arXiv [q-bio.GN]*. Available at: http://arxiv.org/abs/1703.09852.

Góngora-Castillo, E. and Buell, C. R. (2013) 'Bioinformatics challenges in de novo transcriptome assembly using short read sequences in the absence of a reference genome sequence', *Natural Product Reports*, 30(4), pp. 490–500.

Gonnet, G. H. *et al.* (2000) 'Darwin v. 2.0: an interpreted computer language for the biosciences', *Bioinformatics*, 16(2), pp. 101–103.

Gonnet, G. H., Cohen, M. A. and Benner, S. A. (1992) 'Exhaustive matching of the entire protein sequence database', *Science*, 256(5062), pp. 1443–1445.

Goodstein, D. M. *et al.* (2012) 'Phytozome: a comparative platform for green plant genomics', *Nucleic Acids Research*, 40(Database issue), pp. D1178–D1186.

Goodwin, S., McPherson, J. D. and McCombie, W. R. (2016) 'Coming of age: ten years of next-generation sequencing technologies', *Nature Reviews Genetics*, 17(6), pp. 333–351.

Google, LLC (2004) *Google Scholar*. Available at: https://scholar.google.com (Accessed: 7 December 2019).

Gotoh, O. (1999) ‘Multiple sequence alignment: algorithms and applications’, *Advances in Biophysics*, 36, pp. 159–206.

Gotoh, O. (2000) ‘Homology-based gene structure prediction: simplified matching algorithm using a translated codon (tron) and improved accuracy by allowing for long gaps’, *Bioinformatics*, 16(3), pp. 190–202.

Goutte, C. and Gaussier, E. (2005) ‘A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation’, in Losada, D. E. and Fernández-Luna, J. M. (eds) *Advances in Information Retrieval. ECIR 2005. Lecture Notes in Computer Science*. Heidelberg: Springer, Berlin, Heidelberg, pp. 345–359.

Gowik, U. *et al.* (2011) ‘Evolution of C4 photosynthesis in the genus Flaveria: how many and which genes does it take to make C4?’, *The Plant Cell*, 23(6), pp. 2087–2105.

Grabherr, M. G. *et al.* (2011) ‘Full-length transcriptome assembly from RNA-Seq data without a reference genome’, *Nature biotechnology*, 29(7), pp. 644–652.

Gray, K. A. *et al.* (2016) ‘A review of the new HGNC gene family resource’, *Human Genomics*, 10, p. 6.

Gribskov, M., McLachlan, A. D. and Eisenberg, D. (1987) ‘Profile analysis: detection of distantly related proteins’, *Proceedings of the National Academy of Sciences of the United States of America*, 84(13), pp. 4355–4358.

Griffiths, A. J. F., Miller, J. H. and Suzuki, D. T. (2000) *An Introduction to Genetic Analysis*. New York: W. H. Freeman.

Gront, D. *et al.* (2012) ‘BioShell Threader: protein homology detection based on sequence profiles and secondary structure profiles’, *Nucleic Acids Research*, 40(Web Server issue), pp. W257–W262.

Gros-Balthazard, M. *et al.* (2019) ‘Evolutionary transcriptomics reveals the origins of olives and the genomic changes associated with their

domestication', *The Plant Journal*, 100(1), pp. 143–157.

Guindon, S. *et al.* (2010) 'New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0', *Systematic Biology*, 59(3), pp. 307–321.

Guindon, S. and Gascuel, O. (2003) 'A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood', *Systematic Biology*, 52(5), pp. 696–704.

Gurevich, A. *et al.* (2013) 'QUAST: quality assessment tool for genome assemblies', *Bioinformatics*, 29(8), pp. 1072–1075.

Gu, X., Fu, Y. X. and Li, W. H. (1995) 'Maximum likelihood estimation of the heterogeneity of substitution rate among nucleotide sites', *Molecular Biology and Evolution*, 12(4), pp. 546–557.

Haas, B. J. (2015) *Transcriptome Contig Nx and ExN50 stats*, *trinityrnaseq GitHub*. Available at: https://github.com/trinityrnaseq/trinityrnaseq/wiki/Transcriptome-Contig-Nx-and-ExN50-stats.

Haas, B. J. and Papanicolaou, A. (2016) *TransDecoder*. Available at: https://github.com/TransDecoder.

Haas, O. and Simpson, G. (1946) 'Analysis of some phylogenetic terms, with attempts at redefinition', *Proceedings of the American Philosophical Society*, 90(5), pp. 319–349.

Hardison, R. C. (2003) 'Comparative genomics', *PLOS Biology*, 1(2), p. E58.

Harrison, P. M. (2014) 'Computational methods for pseudogene annotation based on sequence homology', *Methods in Molecular Biology*, 1167, pp. 27–39.

Haseloff, J. *et al.* (1984) 'Striking similarities in amino acid sequence among nonstructural proteins encoded by RNA viruses that have dissimilar genomic organization', *Proceedings of the National Academy of Sciences of the*

United States of America, 81(14), pp. 4358–4362.

Hauser, M., Mayer, C. E. and Söding, J. (2013) 'kClust: fast and sensitive clustering of large protein sequence databases', *BMC Bioinformatics*, 14, p. 248.

Hauser, M., Steinegger, M. and Söding, J. (2016) 'MMseqs software suite for fast and deep clustering and searching of large protein sequence sets', *Bioinformatics*, 32(9), pp. 1323–1330.

Hennig, W. (1965) 'Phylogenetic systematics', *Annual Review of Entomology*, 10(1), pp. 97–116.

Higgins, C. F. (1992) 'ABC transporters: from microorganisms to man', *Annual Review of Cell Biology*, 8, pp. 67–113.

Hillis, D. M. (1994) 'Homology in Molecular Biology', in Hall, B. K. (ed.) *Homology: The Hierarchical Basis of Comparative Biology*. San Diego: Academic Press, pp. 339–368.

Hogeweg, P. and Hesper, B. (1984) 'The alignment of sets of sequences and the construction of phyletic trees: an integrated method', *Journal of Molecular Evolution*, 20(2), pp. 175–186.

Hölzer, M. and Marz, M. (2019) 'De novo transcriptome assembly: A comprehensive cross-species comparison of short-read RNA-Seq assemblers', *GigaScience*, 8(5), p. giz039.

Hosmani, P. S. *et al.* (2019) 'A quick guide for student-driven community genome annotation', *PLOS Computational Biology*, 15(4), p. E1006682.

Howe, K. L. *et al.* (2020) 'Ensembl Genomes 2020—enabling non-vertebrate genomic research', *Nucleic Acids Research*, 48(Database issue), pp. D689–D695.

Hrdlickova, R., Toloue, M. and Tian, B. (2017) 'RNA-Seq methods for transcriptome analysis', *Wiley interdisciplinary reviews. RNA*, 8(1), p. e1364.

Huang, S. *et al.* (2012) ‘HaploMerger: reconstructing allelic relationships for polymorphic diploid genome assemblies’, *Genome Research*, 22(8), pp. 1581–1588.

Huang, S., Kang, M. and Xu, A. (2017) ‘HaploMerger2: rebuilding both haploid sub-assemblies from high-heterozygosity diploid genome assembly’, *Bioinformatics*, 33(16), pp. 2577–2579.

Huelsenbeck, J. P. and Ronquist, F. (2001) ‘MRBAYES: Bayesian inference of phylogenetic trees’, *Bioinformatics*, 17(8), pp. 754–755.

Hughey, R. and Krogh, A. (1996) ‘Hidden Markov models for sequence analysis: extension and analysis of the basic method’, *Computer Applications in the Biosciences*, 12(2), pp. 95–107.

Hu, H., Scheben, A. and Edwards, D. (2018) ‘Advances in Integrating Genomics and Bioinformatics in the Plant Breeding Pipeline’, *Agriculture*, 8(6), p. 75.

Hunt, M. *et al.* (2013) ‘REAPR: a universal tool for genome assembly evaluation’, *Genome Biology*, 14(5), p. R47.

Hunt, M. *et al.* (2014) ‘A comprehensive evaluation of assembly scaffolding tools’, *Genome Biology*, 15(3), p. R42.

Illergård, K., Ardell, D. H. and Elofsson, A. (2009) ‘Structure is three to ten times more conserved than sequence—a study of structural response in protein cores’, *Proteins: Structure, Function, and Bioinformatics*, 77(3), pp. 499–508.

Illumina, Inc. (2020) *rRNA & Globin mRNA Removal Kit Selection Guide*, *Illumina*. Available at: https://www.illumina.com/products/selection-tools/rrna-depletion-selection-guide.html (Accessed: 20 January 2020).

Indrischek, H. *et al.* (2016) ‘The paralog-to-contig assignment problem: high quality gene models from fragmented assemblies’, *Algorithms for Molecular Biology*, 11(1), pp. 1–14.

Ingolia, N. T. (2014) ‘Ribosome profiling: new views of translation, from single codons to genome scale’, *Nature Reviews Genetics*, 15(3), pp. 205–213.

International Wheat Genome Sequencing Consortium (IWGSC) (2014) ‘A chromosome-based draft sequence of the hexaploid bread wheat (Triticum aestivum) genome’, *Science*, 345(6194), p. 1251788.

International Wheat Genome Sequencing Consortium (IWGSC) *et al.* (2018) ‘Shifting the limits in wheat research and breeding using a fully annotated reference genome’, *Science*, 361(6403), p. eaar7191.

Jaakkola, T., Diekhans, M. and Haussler, D. (2000) ‘A discriminative framework for detecting remote protein homologies’, *Journal of Computational Biology*, 7(1-2), pp. 95–114.

Jackman, S. D. *et al.* (2017) ‘ABySS 2.0: resource-efficient assembly of large genomes using a Bloom filter’, *Genome Research*, 27(5), pp. 768–777.

Jacoby, E. *et al.* (2006) ‘The 7 TM G-Protein-Coupled Receptor Target Family’, *ChemMedChem*, 1(8), pp. 760–782.

Jacquier, A. (2009) ‘The complex eukaryotic transcriptome: unexpected pervasive transcription and novel small RNAs’, *Nature Reviews Genetics*, 10(12), pp. 833–844.

Jain, M. *et al.* (2018) ‘Nanopore sequencing and assembly of a human genome with ultra-long reads’, *Nature Biotechnology*, 36(4), pp. 338–345.

Jensen, T. H., Jacquier, A. and Libri, D. (2013) ‘Dealing with pervasive transcription’, *Molecular Cell*, 52(4), pp. 473–484.

Jiao, W.-B. and Schneeberger, K. (2017) ‘The impact of third generation genomic technologies on plant genome assembly’, *Current Opinion in Plant Biology*, 36, pp. 64–70.

Jin Lee, D. and Pyo Hong, C. (2019) ‘Transcriptome Atlas by Long-Read RNA Sequencing: Contribution to a Reference Transcriptome’, in

Blumenberg, M. (ed.) *Transcriptome Analysis*. London: IntechOpen.

Jones, D. T., Taylor, W. R. and Thornton, J. M. (1992) 'The rapid generation of mutation data matrices from protein sequences', *Computer Applications in the Biosciences*, 8(3), pp. 275–282.

Jones, S. J. M. (2006) 'Prediction of genomic functional elements', *Annual Review of Genomics and Human Genetics*, 7, pp. 315–338.

Jones, T. *et al.* (2004) 'The diploid genome sequence of Candida albicans', *Proceedings of the National Academy of Sciences of the United States of America*, 101(19), pp. 7329–7334.

Julca, I. *et al.* (2018) 'Phylogenomics of the olive tree (Olea europaea) reveals the relative contribution of ancient allo- and autopolyploidization events', *BMC Biology*, 16(1), p. 15.

Jun, G. *et al.* (2012) 'Detecting and estimating contamination of human DNA samples in sequencing and array-based genotype data', *American Journal of Human Genetics*, 91(5), pp. 839–848.

Jung, H. *et al.* (2019) 'Tools and Strategies for Long-Read Sequencing and De Novo Assembly of Plant Genomes', *Trends in Plant Science*, 24(8), pp. 700–724.

Kajitani, R. *et al.* (2014) 'Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads', *Genome Research*, 24(8), pp. 1384–1395.

Kajitani, R. *et al.* (2019) 'Platanus-allee is a de novo haplotype assembler enabling a comprehensive access to divergent heterozygous regions', *Nature Communications*, 10(1), p. 1702.

Kapranov, P. *et al.* (2007) 'RNA maps reveal new RNA classes and a possible function for pervasive transcription', *Science*, 316(5830), pp. 1484–1488.

Karchin, R. and Hughey, R. (1998) 'Weighting hidden Markov models for

maximum discrimination', *Bioinformatics*, 14(9), pp. 772–782.

Karplus, K., Barrett, C. and Hughey, R. (1998) 'Hidden Markov models for detecting remote protein homologies', *Bioinformatics*, 14(10), pp. 846–856.

Katoh, K. and Standley, D. M. (2013) 'MAFFT multiple sequence alignment software version 7: improvements in performance and usability', *Molecular Biology and Evolution*, 30(4), pp. 772–780.

Katz, Y. *et al.* (2010) 'Analysis and design of RNA sequencing experiments for identifying isoform regulation', *Nature methods*, 7(12), pp. 1009–1015.

Keller, O. *et al.* (2008) 'Scipio: Using protein sequences to determine the precise exon/intron structures of genes and their orthologs in closely related species', *BMC Bioinformatics*, 9(1), p. 278.

Kelley, D. R. and Salzberg, S. L. (2010) 'Detection and correction of false segmental duplications caused by genome mis-assembly', *Genome Biology*, 11(3), p. R28.

Kelley, L. A. and Sternberg, M. J. E. (2009) 'Protein structure prediction on the Web: a case study using the Phyre server', *Nature Protocols*, 4(3), pp. 363–371.

Kent, A. *et al.* (1955) 'Machine literature searching VIII. Operational criteria for designing information retrieval systems', *American Documentation*, 6(2), pp. 93–101.

Kimura, M. (1983) *The Neutral Theory of Molecular Evolution*. Cambridge: Cambridge University Press.

Kirkpatrick, M. (2010) 'How and why chromosome inversions evolve', *PLOS Biology*, 8(9), p. e1000501.

Kitamoto, Y. *et al.* (1994) 'Enterokinase, the initiator of intestinal digestion, is a mosaic protease composed of a distinctive assortment of domains', *Proceedings of the National Academy of Sciences of the United States of America*, 91(16), pp. 7588–7592.

Knight, J. C. (2004) ‘Allele-specific gene expression uncovered’, *Trends in Genetics*, 20(3), pp. 113–116.

Kolata, G. B. (1980) ‘The 1980 Nobel Prize in Chemistry’, *Science*, 210(4472), pp. 887–889.

Kole, C. (ed.) (2011) *Wild Crop Relatives: Genomic and Breeding Resources: Temperate Fruits*. Heidelberg: Springer, Berlin, Heidelberg.

Kolmogorov, A. N. (1933) ‘Sulla Determinazione Empirica di Una Legge di Distribuzione’, *Giornale dell’Istituto Italiano degli Attuari*, 4, pp. 83–91.

Koning-Boucoiran, C. F. S. *et al.* (2015) ‘Using RNA-Seq to assemble a rose transcriptome with more than 13,000 full-length expressed genes and to develop the WagRhSNP 68k Axiom SNP array for rose (Rosa L.)’, *Frontiers in Plant Science*, 6, p. 249.

Koonin, E. V. and Galperin, M. Y. (2010) *Sequence - Evolution - Function: Computational Approaches in Comparative Genomics*. Boston: Kluwer Academic.

Koren, S. *et al.* (2017) ‘Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation’, *Genome Research*, 27(5), pp. 722–736.

Koren, S. *et al.* (2018) ‘De novo assembly of haplotype-resolved genomes with trio binning’, *Nature Biotechnology*, 36, pp. 1174–1182.

Koren, S., Treangen, T. J. and Pop, M. (2011) ‘Bambus 2: scaffolding metagenomes’, *Bioinformatics*, 27(21), pp. 2964–2971.

Korf, I. (2004) ‘Gene finding in novel genomes’, *BMC Bioinformatics*, 5, p. 59.

Kozarewa, I. *et al.* (2009) ‘Amplification-free Illumina sequencing-library preparation facilitates improved mapping and assembly of (G+C)-biased genomes’, *Nature Methods*, 6(4), pp. 291–295.

Kremer, F. S., McBride, A. J. A. and Pinto, L. da S. (2017) 'Approaches for in silico finishing of microbial genome sequences', *Genetics and Molecular Biology*, 40(3), pp. 553–576.

Krogh, A. *et al.* (1994) 'Hidden Markov models in computational biology. Applications to protein modeling', *Journal of Molecular Biology*, 235(5), pp. 1501–1531.

Krogh, A. (1998) 'An introduction to hidden Markov models for biological sequences', in Salzberg, S. L., Searls, D. B., and Kasif, S. (eds) *Computational Methods in Molecular Biology*. Amsterdam: Elsevier Science B.V., pp. 45–63.

Krogh, A. and Mitchison, G. (1995) 'Maximum entropy weighting of aligned sequences of proteins or DNA', in. *International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 215–221.

Krueger, F. (no date) *Trim Galore!* Available at: http://www.bioinformatics.babraham.ac.uk/projects/trim_galore.

Kukurba, K. R. and Montgomery, S. B. (2015) 'RNA Sequencing and Analysis', *Cold Spring Harbor Protocols*, 2015(11), pp. 951–969.

Kuon, J.-E. *et al.* (2019) 'Haplotype-resolved genomes of geminivirus-resistant and geminivirus-susceptible African cassava cultivars', *BMC Biology*, 17(1), p. 75.

Kyriakidou, M. *et al.* (2018) 'Current Strategies of Polyploid Plant Genome Sequence Assembly', *Frontiers in Plant Science*, 9, p. 1660.

Laehnemann, D., Borkhardt, A. and McHardy, A. C. (2016) 'Denoising DNA deep sequencing data—high-throughput sequencing errors and their correction', *Briefings in Bioinformatics*, 17(1), pp. 154–179.

Lamesch, P. *et al.* (2011) 'The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools', *Nucleic Acids Research*, 40(Database issue), pp. D1202–D1210.

Lander, E. S. *et al.* (2001) ‘Initial sequencing and analysis of the human genome’, *Nature*, 409(6822), pp. 860–921.

Langmead, B. and Salzberg, S. L. (2012) ‘Fast gapped-read alignment with Bowtie 2’, *Nature Methods*, 9(4), pp. 357–359.

Lankester, E. R. (1870) ‘On the use of the term homology in modern zoology, and the distinction between homogenetic and homoplastic agreement’, *The Annals and Magazine of Natural History*. (4), 6, pp. 34–43.

Laplace, P. S. de (1836) *Théorie analytique des probabilités*. 3rd edn. Paris: Courcier.

Larsson, A. (2014) ‘AliView: a fast and lightweight alignment viewer and editor for large datasets’, *Bioinformatics*, 30(22), pp. 3276–3278.

Lartillot, N. and Philippe, H. (2004) ‘A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process’, *Molecular Biology and Evolution*, 21(6), pp. 1095–1109.

Lee, C., Grasso, C. and Sharlow, M. F. (2002) ‘Multiple sequence alignment using partial order graphs’, *Bioinformatics*, 18(3), pp. 452–464.

Lee, H. and Tang, H. (2012) ‘Next-generation sequencing technologies and fragment assembly algorithms’, *Methods in molecular biology* , 855, pp. 155–174.

Le, S. Q. and Gascuel, O. (2008) ‘An improved general amino acid replacement matrix’, *Molecular Biology and Evolution*, 25(7), pp. 1307–1320.

Levin, J. Z. *et al.* (2010) ‘Comprehensive comparative analysis of strand-specific RNA sequencing methods’, *Nature Methods*, 7(9), pp. 709–715.

Levy, S. *et al.* (2007) ‘The diploid genome sequence of an individual human’, *PLOS Biology*, 5(10), p. e254.

Lewis, F., Butler, A. and Gilbert, L. (2011) ‘A unified approach to model selection using the likelihood ratio test’, *Methods in Ecology and Evolution*,

2(2), pp. 155–162.

Liang, C. *et al.* (2009) ‘Evidence-based gene predictions in plant genomes’, *Genome Research*, 19(10), pp. 1912–1923.

Lima, L. *et al.* (2017) ‘Playing hide and seek with repeats in local and global de novo transcriptome assembly of short RNA-seq reads’, *Algorithms for Molecular Biology*, 12, p. 2.

Li, R. *et al.* (2010) ‘The sequence and de novo assembly of the giant panda genome’, *Nature*, 463(7279), pp. 311–317.

Liu, B., Chen, J. and Wang, X. (2015) ‘Application of learning to rank to protein remote homology detection’, *Bioinformatics* , 31(21), pp. 3492–3498.

Liu, M. *et al.* (2014) ‘A Transcriptome Post-Scaffolding Method for Assembling High Quality Contigs’, *Computational Biology Journal*, 2014, p. 961823.

Li, W. and Godzik, A. (2006) ‘Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences’, *Bioinformatics*, 22(13), pp. 1658–1659.

Li, Y. I. and Copley, R. R. (2013) ‘Scaffolding low quality genomes using orthologous protein sequences’, *Bioinformatics*, 29(2), pp. 160–165.

Li, Z. *et al.* (2012) ‘Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph’, *Briefings in Functional Genomics*, 11(1), pp. 25–37.

Lodish, H. *et al.* (eds) (2000) ‘Mutations: types and causes’, in *Molecular Cell Biology*. New York: W. H. Freeman.

Lomsadze, A. *et al.* (2005) ‘Gene identification in novel eukaryotic genomes by self-training algorithm’, *Nucleic Acids Research*, 33(20), pp. 6494–6506.

Loureiro, J. *et al.* (2007) ‘Nuclear DNA content estimations in wild olive (Olea europaea L. ssp. europaea var. sylvestris Brot.) and Portuguese cultivars of

O. europaea using flow cytometry', *Genetic Resources and Crop Evolution*, 54(1), pp. 21–25.

Löytynoja, A. (2014) 'Phylogeny-aware alignment with PRANK', *Methods in Molecular Biology* , 1079, pp. 155–170.

Löytynoja, A., Vilella, A. J. and Goldman, N. (2012) 'Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm', *Bioinformatics*, 28(13), pp. 1684–1691.

Lu, B., Zeng, Z. and Shi, T. (2013) 'Comparative study of de novo assembly and genome-guided assembly strategies for transcriptome reconstruction based on RNA-Seq', *Science China Life Sciences*, 56(2), pp. 143–155.

Luczak, B. B., James, B. T. and Girgis, H. Z. (2019) 'A survey and evaluations of histogram-based statistics in alignment-free sequence comparison', *Briefings in Bioinformatics*, 20(4), pp. 1222–1237.

Lybecker, M., Bilusic, I. and Raghavan, R. (2014) 'Pervasive transcription: detecting functional RNAs in bacteria', *Transcription*, 5(4), p. e944039.

Lynch, M. and Conery, J. S. (2000) 'The evolutionary fate and consequences of duplicate genes', *Science*, 290(5494), pp. 1151–1155.

Mandric, I., Knyazev, S. and Zelikovsky, A. (2018) 'Repeat-aware evaluation of scaffolding tools', *Bioinformatics*, 34(15), pp. 2530–2537.

Manzoni, C. *et al.* (2018) 'Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences', *Briefings in Bioinformatics*, 19(2), pp. 286–302.

Marchler-Bauer, A. *et al.* (2016) 'CDD/SPARCLE: functional classification of proteins via subfamily domain architectures', *Nucleic Acids Research*, 45(Database issue), pp. D200–D203.

Marcussen, T. *et al.* (2014) 'Ancient hybridizations among the ancestral genomes of bread wheat', *Science*, 345(6194), p. 1250092.

Mardis, E. R. (2017) ‘DNA sequencing technologies: 2006-2016’, *Nature Protocols*, 12(2), pp. 213–218.

Margelevičius, M. and Venclovas, Č. (2010) ‘Detection of distant evolutionary relationships between protein families using theory of sequence profile-profile comparison’, *BMC Bioinformatics*, 11(1), p. 89.

Margulies, E. H. and Birney, E. (2008) ‘Approaches to comparative sequence analysis: towards a functional view of vertebrate genomes’, *Nature Reviews Genetics*, 9(4), pp. 303–313.

Margulies, M. *et al.* (2005) ‘Genome sequencing in microfabricated high-density picolitre reactors’, *Nature*, 437(7057), pp. 376–380.

Martin, J. A. and Wang, Z. (2011) ‘Next-generation transcriptome assembly’, *Nature Reviews Genetics*, 12(10), pp. 671–682.

Masel, J. (2011) ‘Genetic drift’, *Current Biology*, 21(20), pp. R837–R838.

Maxam, A. M. and Gilbert, W. (1977) ‘A new method for sequencing DNA’, *Proceedings of the National Academy of Sciences of the United States of America*, 74(2), pp. 560–564.

McCarthy, A. (2010) ‘Third generation DNA sequencing: pacific biosciences’ single molecule real time technology’, *Chemistry & Biology*, 17(7), pp. 675–676.

McGeoch, D. J. and Davison, A. J. (1986) ‘DNA sequence of the herpes simplex virus type 1 gene encoding glycoprotein gH, and identification of homologues in the genomes of varicella-zoster virus and Epstein-Barr virus’, *Nucleic Acids Research*, 14(10), pp. 4281–4292.

Mehrotra, S. and Goyal, V. (2014) ‘Repetitive sequences in plant nuclear DNA: types, distribution, evolution and function’, *Genomics, Proteomics & Bioinformatics*, 12(4), pp. 164–171.

Meltz Steinberg, K. *et al.* (2017) ‘Building and Improving Reference Genome Assemblies’, *Proceedings of the IEEE*, 105(3), pp. 422–435.

Messing, J., Crea, R. and Seeburg, P. H. (1981) ‘A system for shotgun DNA sequencing’, *Nucleic Acids Research*, 9(2), pp. 309–321.

Metzker, M. L. (2009) ‘Sequencing technologies—the next generation’, *Nature Reviews Genetics*, 11(1), pp. 31–46.

Mi, H. *et al.* (2019) ‘PANTHER version 14: more genomes, a new PANTHER GO-slim and improvements in enrichment analysis tools’, *Nucleic Acids Research*, 47(Database issue), pp. D419–D426.

Miller, W. *et al.* (2004) ‘Comparative genomics’, *Annual Review of Genomics and Human Genetics*, 5, pp. 15–56.

Mira, A. *et al.* (2010) ‘The bacterial pan-genome: a new paradigm in microbiology’, *International Microbiology*, 13(2), pp. 45–57.

Morrison, D. A., Morgan, M. J. and Kelchner, S. A. (2015) ‘Molecular homology and multiple-sequence alignment: an analysis of concepts and practice’, *Australian Systematic Botany*, 28(1), pp. 46–62.

Mortazavi, A. *et al.* (2010) ‘Scaffolding a Caenorhabditis nematode genome with RNA-seq’, *Genome Research*, 20(12), pp. 1740–1747.

Muggli, M. D. *et al.* (2015) ‘Misassembly detection using paired-end sequence reads and optical mapping data’, *Bioinformatics*, 31(12), pp. i80–i88.

Mukherjee, S. *et al.* (2017) ‘Genomes OnLine Database (GOLD) v.6: data updates and feature enhancements’, *Nucleic Acids Research*, 45(Database issue), pp. D446–D456.

Mukherjee, S. *et al.* (2019) ‘Genomes OnLine database (GOLD) v.7: updates and new features’, *Nucleic Acids Research*, 47(Database issue), pp. D649–D659.

Müller, S. *et al.* (2012) ‘Comparison of transcriptome technologies in the pathogenic fungus Aspergillus fumigatus reveals novel insights into the genome and MpkA dependent gene expression’, *BMC Genomics*, 13(1), p.

519.

Mun, T. *et al.* (2016) ‘Lotus Base: An integrated information portal for the model legume Lotus japonicus’, *Scientific Reports*, 6, p. 39447.

Munoz-Merida, A. *et al.* (2013) ‘De Novo Assembly and Functional Annotation of the Olive (Olea europaea) Transcriptome’, *DNA Research*, 20(1), pp. 93–108.

Mutwil, M. *et al.* (2010) ‘Assembly of an interactive correlation network for the Arabidopsis genome using a novel heuristic clustering algorithm’, *Plant Physiology*, 152(1), pp. 29–43.

Nagarajan, N. and Pop, M. (2013) ‘Sequence assembly demystified’, *Nature Reviews Genetics*, 14(3), pp. 157–167.

Nakanishi, S. (1992) ‘Molecular diversity of glutamate receptors and implications for brain function’, *Science*, 258(5082), pp. 597–603.

NCBI Resource Coordinators (2017) ‘Database Resources of the National Center for Biotechnology Information’, *Nucleic Acids Research*, 45(Database issue), pp. D12–D17.

Needleman, S. B. (1970) ‘Needleman-Wunsch algorithm for sequence similarity searches’, *Journal of Molecular Biology*, 48, pp. 443–453.

Nei, M., Suzuki, Y. and Nozawa, M. (2010) ‘The neutral theory of molecular evolution in the genomic era’, *Annual review of genomics and human genetics*, 11, pp. 265–289.

Neri, F. *et al.* (2017) ‘Intragenic DNA methylation prevents spurious transcription initiation’, *Nature*, 543(7643), pp. 72–77.

Nilsen, T. W. and Graveley, B. R. (2010) ‘Expansion of the eukaryotic proteome by alternative splicing’, *Nature*, 463(7280), pp. 457–463.

Notredame, C., Higgins, D. G. and Heringa, J. (2000) ‘T-Coffee: A novel method for fast and accurate multiple sequence alignment’, *Journal of*

Molecular Biology, 302(1), pp. 205–217.

Obayashi, T. *et al.* (2018) 'ATTED-II in 2018: A Plant Coexpression Database Based on Investigation of the Statistical Property of the Mutual Rank Index', *Plant & Cell Physiology*, 59(2), p. 440.

Ocaña, K. and de Oliveira, D. (2015) 'Parallel computing in genomic research: advances and applications', *Advances and Applications in Bioinformatics and Chemistry*, 8, pp. 23–35.

Ohta, T. (2000) 'Evolution of gene families', *Gene*, 259(1-2), pp. 45–52.

Ojeda, D. I. *et al.* (2019) 'Utilization of Tissue Ploidy Level Variation in de Novo Transcriptome Assembly of Pinus sylvestris', *G3*, 9(10), pp. 3409–3421.

O'Neil, S. T. and Emrich, S. J. (2013) 'Assessing De Novo transcriptome assembly metrics for consistency and utility', *BMC Genomics*, 14, p. 465.

Otto, S. P. (2007) 'The evolutionary consequences of polyploidy', *Cell*, 131(3), pp. 452–462.

Otto, T. D. *et al.* (2011) 'RATT: Rapid Annotation Transfer Tool', *Nucleic Acids Research*, 39(9), p. e57

Owen, R. (1843) *Lectures on the comparative anatomy and physiology of the invertebrate animals, delivered at the Royal College of Surgeons*. London: Longman, Brown, Green, and Longmans.

Palazzo, A. F. and Lee, E. S. (2015) 'Non-coding RNA: what is functional and what is junk?', *Frontiers in Genetics*. Frontiers, 6, p. 2.

Pan, Q. *et al.* (2008) 'Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing', *Nature Genetics*, 40(12), pp. 1413–1415.

Park, S.-G. *et al.* (2017) 'Long-read transcriptome data for improved gene prediction in Lentinula edodes', *Data in Brief*, 15, pp. 454–458.

Parra, G., Bradnam, K. and Korf, I. (2007) ‘CEGMA: a pipeline to accurately annotate core genes in eukaryotic genomes’, *Bioinformatics*, 23(9), pp. 1061–1067.

Patterson, C. (1988) ‘Homology in classical and molecular biology’, *Molecular Biology and Evolution*, 5(6), pp. 603–625.

Paulino, D. *et al.* (2015) ‘Sealer: a scalable gap-closing application for finishing draft genomes’, *BMC Bioinformatics*, 16, p. 230.

Paux, E. *et al.* (2008) ‘A physical map of the 1-gigabase bread wheat chromosome 3B’, *Science*, 322(5898), pp. 101–104.

Pearson, W. R. (2000) ‘Flexible sequence similarity searching with the FASTA3 program package’, *Methods in Molecular Biology*, 132, pp. 185–219.

Pearson, W. R. (2013) ‘An Introduction to Sequence Similarity (“Homology”) Searching’, *Current Protocols in Bioinformatics*, 42(1), pp. 3.1.1–3.1.8.

Pearson, W. R. and Lipman, D. J. (1988) ‘Improved tools for biological sequence comparison’, *Proceedings of the National Academy of Sciences of the United States of America*, 85(8), pp. 2444–2448.

Pearson, W. R. and Sierk, M. L. (2005) ‘The limits of protein sequence comparison?’, *Current Opinion in Structural Biology*, 15(3), pp. 254–260.

Pereira, M. A., Imada, E. L. and Guedes, R. L. M. (2017) ‘RNA□seq: Applications and Best Practices’, in Marchi, F., Cirillo, P., and Mateo, E. C. (eds) *Applications of RNA-Seq and Omics Strategies*. London: IntechOpen.

Pesaran, M. H. and Weeks, M. (1999) ‘Non-nested hypothesis testing: an overview’, *Cambridge Working Papers in Economics*. Faculty of Economics, University of Cambridge.

Pfeifer, M. *et al.* (2014) ‘Genome interplay in the grain transcriptome of hexaploid bread wheat’, *Science*, 345(6194), p. 1250091.

Pfeil, B. E. *et al.* (2005) ‘Placing paleopolyploidy in relation to taxon divergence: a phylogenetic analysis in legumes using 39 gene families’, *Systematic Biology*, 54(3), pp. 441–454.

Phillippy, A. M., Schatz, M. C. and Pop, M. (2008) ‘Genome assembly forensics: finding the elusive mis-assembly’, *Genome Biology*, 9(3), p. R55..

Phoenix Bioinformatics Corporation (2010) *Genome Assembly*, *The Arabidopsis Information Resource (TAIR)*. Available at: https://www.arabidopsis.org/portals/genAnnotation/gene_structural_annotation/agicomplete.jsp (Accessed: 20 July 2019).

Phytozome (2017) *Manihot esculenta v6.1 (Cassava).* Available at: https://phytozome.jgi.doe.gov/Mesculenta (Accessed: 26 October 2017).

Piližota, I. *et al.* (2019) ‘Phylogenetic approaches to identifying fragments of the same gene, with application to the wheat genome’, *Bioinformatics*, 35(7), pp. 1159–1166.

Pinter, S. F. *et al.* (2015) ‘Allelic Imbalance Is a Prevalent and Tissue-Specific Feature of the Mouse Transcriptome’, *Genetics*, 200(2), pp. 537–549.

Pollard, M. O. *et al.* (2018) ‘Long reads: their purpose and place’, *Human Molecular Genetics*, 27(R2), pp. R234–R241.

Pop, M. (2009) ‘Genome assembly reborn: recent computational challenges’, *Briefings in Bioinformatics*, 10(4), pp. 354–366.

Portin, P. and Wilkins, A. (2017) ‘The Evolving Definition of the Term “Gene”’, *Genetics*, 205(4), pp. 1353–1364.

Price, M. N., Dehal, P. S. and Arkin, A. P. (2009) ‘FastTree: computing large minimum evolution trees with profiles instead of a distance matrix’, *Molecular Biology and Evolution*, 26(7), pp. 1641–1650.

Price, M. N., Dehal, P. S. and Arkin, A. P. (2010) ‘FastTree 2 –

Approximately Maximum-Likelihood Trees for Large Alignments', *PLOS ONE*, 5(3), p. e9490.

Prochnik, S. *et al.* (2012) 'The Cassava Genome: Current Progress, Future Directions', *Tropical Plant Biology*, 5(1), pp. 88–94.

Proudfoot, N. J., Furger, A. and Dye, M. J. (2002) 'Integrating mRNA processing with transcription', *Cell*, 108(4), pp. 501–512.

Pryszcz, L. P. and Gabaldón, T. (2016) 'Redundans: an assembly pipeline for highly heterozygous genomes', *Nucleic Acids Research*, 44(12), p. e113.

Rabiner, L. R. and Juang, B.-H. (1986) 'An introduction to hidden Markov models', *IEEE ASSP Magazine*, 3(1), pp. 4–16.

R Core Team (2017) *R: A language and environment for statistical computing*. Available at: https://www.R-project.org.

Remmert, M. *et al.* (2011) 'HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment', *Nature Methods*, 9(2), pp. 173–175.

Renaut, S. *et al.* (2018) 'Genome Survey of the Freshwater Mussel Venustaconcha ellipsiformis (Bivalvia: Unionida) Using a Hybrid De Novo Assembly Approach', *Genome Biology and Evolution*, 10(7), pp. 1637–1646.

Reyes, A. and Huber, W. (2018) 'Alternative start and termination sites of transcription drive most transcript isoform differences across human tissues', *Nucleic Acids Research*, 46(2), pp. 582–592.

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

Rice, E. S. and Green, R. E. (2019) 'New Approaches for Genome Assembly and Scaffolding', *Annual Review of Animal Biosciences*, 7, pp. 17–40.

Richardson, E. J. and Watson, M. (2013) 'The automatic annotation of bacterial genomes', *Briefings in Bioinformatics*, 14(1), pp. 1–12.

Richards, S. (2018) 'Full disclosure: Genome assembly is still hard', *PLOS*

Biology, p. e2005894.

Rigden, D. J. (2017) *From Protein Structure to Function with Bioinformatics*. Dordrecht: Springer.

Riley, F. R. (2002) 'Olive oil production on bronze age Crete: nutritional properties, processing methods and storage life of Minoan olive oil', *Oxford Journal of Archaeology*, 21(1), pp. 63–75.

Rimbert, H. *et al.* (2018) 'High throughput SNP discovery and genotyping in hexaploid wheat', *PLOS ONE*, 13(1), p. e0186329.

Robinson, O., Dylus, D. and Dessimoz, C. (2016) 'Phylo.io: Interactive Viewing and Comparison of Large Phylogenetic Trees on the Web', *Molecular Biology and Evolution*, 33(8), pp. 2163–2166.

Rodríguez, F. *et al.* (1990) 'The general stochastic model of nucleotide substitution', *Journal of Theoretical Biology*, 142(4), pp. 485–501.

Roth, A. C. J., Gonnet, G. H. and Dessimoz, C. (2008) 'Algorithm of OMA for large-scale orthology inference', *BMC Bioinformatics*, 9, p. 518.

Roukos, V., Burman, B. and Misteli, T. (2013) 'The cellular etiology of chromosome translocations', *Current Opinion in Cell Biology*, 25(3), pp. 357–364.

Rugini, E. *et al.* (2011) 'Olea', in Kole, C. (ed.) *Wild Crop Relatives: Genomic and Breeding Resources: Temperate Fruits*. Heidelberg: Springer Berlin Heidelberg, pp. 79–117.

Ruttink, T. *et al.* (2013) 'Orthology Guided Assembly in highly heterozygous crops: creating a reference transcriptome to uncover genetic diversity in Lolium perenne', *Plant Biotechnology Journal*, 11(5), pp. 605–617.

Rychlewski, L. *et al.* (2000) 'Comparison of sequence profiles. Strategies for structural predictions using sequence information', *Protein Science*, 9(2), pp. 232–241.

Rzhetsky, A. and Nei, M. (1992) ‘A simple method for estimating and testing minimum-evolution trees’, *Molecular Biology and Evolution*, 9(5), p. 945.

Sadreyev, R. and Grishin, N. (2003) ‘COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance’, *Journal of Molecular Biology*, 326(1), pp. 317–336.

Šafář, J. *et al.* (2010) ‘Development of Chromosome-Specific BAC Resources for Genomics of Bread Wheat’, *Cytogenetic and Genome Research*, 129(1-3), pp. 211–223.

Safonova, Y., Bankevich, A. and Pevzner, P. A. (2014) ‘dipSPAdes: Assembler for Highly Polymorphic Diploid Genomes’, in Sharan, R. (ed.) *Research in Computational Molecular Biology. RECOMB 2014. Lecture Notes in Computer Science*. Cham: Springer, pp. 265–279.

Saha, S. *et al.* (2008) ‘Empirical comparison of ab initio repeat finding programs’, *Nucleic Acids Research*, 36(7), pp. 2284–2294.

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

Salzberg, S. L. (2019) ‘Next-generation genome annotation: we still struggle to get it right’, *Genome Biology*, 20(1), p. 92.

Salzberg, S. L. and Yorke, J. A. (2005) ‘Beware of mis-assembled genomes’, *Bioinformatics*, 21(24), pp. 4320–4321.

Sam, L. T. *et al.* (2011) ‘A comparison of single molecule and amplification based sequencing of cancer transcriptomes’, *PLOS ONE*, 6(3), p. e17305.

Sanderson, M. J. and Hufford, L. (1996) *Homoplasy: The Recurrence of Similarity in Evolution*. London: Academic Press.

Sanger, F. and Coulson, A. R. (1975) ‘A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase’, *Journal of Molecular Biology*, 94(3), pp. 441–448.

Sanger, F., Nicklen, S. and Coulson, A. R. (1977) ‘DNA sequencing with chain-terminating inhibitors’, *Proceedings of the National Academy of Sciences of the United States of America*, 74(12), pp. 5463–5467.

Sangiovanni, M. *et al.* (2019) ‘From trash to treasure: detecting unexpected contamination in unmapped NGS data’, *BMC Bioinformatics*, 20(Suppl 4), p. 168.

Sayers, E. W. *et al.* (2019) ‘Database resources of the National Center for Biotechnology Information’, *Nucleic Acids Research*, 47(Database issue), pp. D23–D28.

Sayre, R. *et al.* (2011) ‘The BioCassava plus program: biofortification of cassava for sub-Saharan Africa’, *Annual Review of Plant Biology*, 62, pp. 251–272.

Sayyari, E., Whitfield, J. B. and Mirarab, S. (2017) ‘Fragmentary Gene Sequences Negatively Impact Gene Tree and Species Tree Reconstruction’, *Molecular Biology and Evolution*, 34(12), pp. 3279–3291.

Schäffer, A. A. *et al.* (1999) ‘IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices’, *Bioinformatics*, 15(12), pp. 1000–1011.

Scheben, A., Yuan, Y. and Edwards, D. (2016) ‘Advances in genomics for adapting crops to climate change’, *Current Plant Biology*, 6, pp. 2–10.

Schliesky, S. *et al.* (2012) ‘RNA-Seq Assembly – Are We There Yet?’, *Frontiers in plant science*, 3, p. 220.

Schlueter, J. A. *et al.* (2007) ‘Gene duplication and paleopolyploidy in soybean and the implications for whole genome sequencing’, *BMC Genomics*, 8, p. 330.

Schmieder, R. and Edwards, R. (2011) ‘Fast identification and removal of sequence contamination from genomic and metagenomic datasets’, *PLOS ONE*, 6(3), p. e17288.

Schnable, P. S. *et al.* (2009) 'The B73 maize genome: complexity, diversity, and dynamics', *Science*, 326(5956), pp. 1112–1115.

Schneeberger, K. *et al.* (2011) 'Reference-guided assembly of four diverse Arabidopsis thaliana genomes', *Proceedings of the National Academy of Sciences of the United States of America*, 108(25), pp. 10249–10254.

Sedlazeck, F. J. *et al.* (2018) 'Piercing the dark matter: bioinformatics of long-range sequencing and mapping', *Nature Reviews Genetics*, 19(6), pp. 329–346.

Shaffer, C. (2007) 'Next-generation sequencing outpaces expectations', *Nature Biotechnology*, 25(2), p. 149.

Sharan, R., Ulitsky, I. and Shamir, R. (2007) 'Network□based prediction of protein function', *Molecular Systems Biology*, 3(1), p. 88.

Sharon, D. *et al.* (2013) 'A single-molecule long-read survey of the human transcriptome', *Nature Biotechnology*, 31(11), pp. 1009–1014.

Shendure, J. and Ji, H. (2008) 'Next-generation DNA sequencing', *Nature Biotechnology*, 26(10), p. nbt1486.

Shimodaira, H. and Hasegawa, M. (1999) 'Multiple comparisons of log-likelihoods with applications to phylogenetic inference', *Molecular Biology and Evolution*, 16(8), p. 1114.

Shlyueva, D., Stampfel, G. and Stark, A. (2014) 'Transcriptional enhancers: from properties to genome-wide predictions', *Nature Reviews Genetics*, 15(4), pp. 272–286.

Sievers, F. *et al.* (2011) 'Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega', *Molecular Systems Biology*, 7, p. 539.

Silva, G. G. *et al.* (2013) 'Combining de novo and reference-guided assembly with scaffold_builder', *Source Code for Biology and Medicine*, 8(1), p. 23.

Simão, F. A. *et al.* (2015) 'BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs', *Bioinformatics*, 31(19), pp. 3210–3212.

Simpson, J. T. *et al.* (2009) 'ABySS: a parallel assembler for short read sequence data', *Genome Research*, 19(6), pp. 1117–1123.

Simpson, J. T. and Durbin, R. (2012) 'Efficient de novo assembly of large genomes using compressed data structures', *Genome Research*, 22(3), pp. 549–556.

Sleator, R. D. (2010) 'An overview of the current status of eukaryote gene prediction strategies', *Gene*, 461(1-2), pp. 1–4.

Smirnov, N. (1948) 'Table for Estimating the Goodness of Fit of Empirical Distributions', *Annals of Mathematical Statistics*, 19(2), pp. 279–281.

Smit, A. F. A., Hubley, R. and Green, R. (2013-2015) *RepeatMasker Open-4.0*. Available at: http://www.repeatmasker.org.

Smith, T. F. and Waterman, M. S. (1981) 'Identification of common molecular subsequences', *Journal of Molecular Biology*, 147(1), pp. 195–197.

Smith-Unna, R. *et al.* (2016) 'TransRate: reference-free quality assessment of de novo transcriptome assemblies', *Genome research*, 26(8), pp. 1134–1144.

Snyder, M. W. *et al.* (2015) 'Haplotype-resolved genome sequencing: experimental methods and applications', *Nature Reviews Genetics*, 16(6), pp. 344–358.

Sohn, J.-I. and Nam, J.-W. (2016) 'The present and future of de novo whole-genome assembly', *Briefings in Bioinformatics*, 19(1), pp. 23–40.

Soltis, D. E. and Soltis, P. S. (2003) 'The role of phylogenetics in comparative genetics', *Plant Physiology*, 132(4), pp. 1790–1800.

Song, L. and Florea, L. (2015) 'Rcorrector: efficient and accurate error

correction for Illumina RNA-seq reads', *GigaScience*, 4, p. 48.

Song, N. *et al.* (2008) 'Sequence similarity network reveals common ancestry of multidomain proteins', *PLOS Computational Biology*, 4(4), p. e1000063.

Stamatakis, A. (2006) 'Phylogenetic models of rate heterogeneity: a high performance computing perspective', in *Proceedings 20th IEEE International Parallel Distributed Processing Symposium*, p. 8.

Stamatakis, A. (2006) 'RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models', *Bioinformatics*, 22(21), pp. 2688–2690.

Stamatakis, A. (2014) 'RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies', *Bioinformatics*, 30(9), pp. 1312–1313.

Stamatakis, A. (2016) *The RAxML v8. 2. X Manual*. Heidelberg Institute for Theoretical Studies. Available at: https://sco.h-its.org/exelixis/resource/download/NewManual.pdf.

Steijger, T. *et al.* (2013) 'Assessment of transcript reconstruction methods for RNA-seq', *Nature Methods*, 10(12), pp. 1177–1184.

Steinegger, M. *et al.* (2019) 'HH-suite3 for fast remote homology detection and deep protein annotation', *BMC Bioinformatics*, 20(1), p. 473.

Steinegger, M. and Söding, J. (2017) 'MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets', *Nature Biotechnology*, 35(11), pp. 1026–1028.

Stelzl, U. *et al.* (2005) 'A human protein-protein interaction network: a resource for annotating the proteome', *Cell*, 122(6), pp. 957–968.

Stephens, M. (2017) *The Beta Distribution*, *fiveMinuteStats*. Available at: https://stephens999.github.io/fiveMinuteStats/beta.html (Accessed: 14 January 2020).

Steuernagel, B. *et al.* (2009) 'De novo 454 sequencing of barcoded BAC

pools for comprehensive gene survey and genome analysis in the complex genome of barley', *BMC Genomics*, 10, p. 547.

Stočes, Š. *et al.* (2016) 'Orthology Guided Transcriptome Assembly of Italian Ryegrass and Meadow Fescue for Single-Nucleotide Polymorphism Discovery', *The Plant Genome*, 9(3), pp. 1–14.

Suarez-Gonzalez, A., Lexer, C. and Cronk, Q. C. B. (2018) 'Adaptive introgression: a plant perspective', *Biology Letters*, 14(3).

Sun, Z. *et al.* (2013) 'Impact of library preparation on downstream analysis and interpretation of RNA-Seq data: comparison between Illumina PolyA and NuGEN Ovation protocol', *PLOS ONE*, 8(8), p. e71745.

Swofford, D. L. (2000) *PAUP*: Phylogenetic Analysis by Parsimony (and Other Methods) 4.0*.

Szalkowski, A. *et al.* (2008) 'SWPS3 - fast multi-threaded vectorized Smith-Waterman for IBM Cell/B.E. and x86/SSE2', *BMC Research Notes*, 1, p. 107.

Szalkowski, A. M. (2012) 'Fast and robust multiple sequence alignment with phylogeny-aware gap placement', *BMC Bioinformatics*, 13, p. 129.

Szymanski, M. *et al.* (2016) '5SRNAdb: an information resource for 5S ribosomal RNAs', *Nucleic Acids Research*, 44(Database issue), pp. D180–D183.

Tamura, K. *et al.* (2011) 'MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods', *Molecular Biology and Evolution*, 28(10), pp. 2731–2739.

Tardaguila, M. *et al.* (2018) 'SQANTI: extensive characterization of long-read transcript sequences for quality control in full-length transcriptome identification and quantification', *Genome Research*, 28(3), pp. 396–411.

Tatusov, R. L. *et al.* (1996) 'Metabolism and evolution of Haemophilus influenzae deduced from a whole-genome comparison with Escherichia coli', *Current Biology*, 6(3), pp. 279–291.

Tatusov, R. L. *et al.* (2003) ‘The COG database: an updated version includes eukaryotes’, *BMC Bioinformatics*, 4, p. 41.

Tatusov, R. L., Koonin, E. V. and Lipman, D. J. (1997) ‘A genomic perspective on protein families’, *Science*, 278(5338), pp. 631–637.

Taylor, J. S. and Raes, J. (2004) ‘Duplication and divergence: the evolution of new genes and old ideas’, *Annual Review of Genetics*, 38, pp. 615–643.

Tcherepanov, V., Ehlers, A. and Upton, C. (2006) ‘Genome Annotation Transfer Utility (GATU): rapid annotation of viral genomes using a closely related reference genome’, *BMC Genomics*, 7, p. 150.

Ter-Hovhannisyan, V. *et al.* (2008) ‘Gene prediction in novel fungal genomes using an ab initio algorithm with unsupervised training’, *Genome Research*, 18(12), pp. 1979–1990.

The 1000 Genomes Project Consortium (2015) ‘A global reference for human genetic variation’, *Nature*, 526(7571), pp. 68–74.

The Arabidopsis Genome Initiative (2000) ‘Analysis of the genome sequence of the flowering plant Arabidopsis thaliana’, *Nature*, 408(6814), pp. 796–815.

Thomas, P. D. *et al.* (2003) ‘PANTHER: a library of protein families and subfamilies indexed by function’, *Genome Research*, 13(9), pp. 2129–2141.

Thomma, B. P. H. J. *et al.* (2016) ‘Mind the gap; seven reasons to close fragmented genome assemblies’, *Fungal Genetics and Biology*, 90, pp. 24–30.

Tischler-Höhle, G. (2019) ‘Haplotype and Repeat Separation in Long Reads’, in Bartoletti, M. et al. (eds) *Computational Intelligence Methods for Bioinformatics and Biostatistics. CIBB 2017. Lecture Notes in Computer Science*. Cham: Springer, pp. 103–114.

Tohge, T. and Fernie, A. R. (2012) ‘Co-expression and co-responses: within and beyond transcription’, *Frontiers in Plant Science*, 3, p. 248.

Toh, H., Hayashida, H. and Miyata, T. (1983) ‘Sequence homology between retroviral reverse transcriptase and putative polymerases of hepatitis B virus and cauliflower mosaic virus’, *Nature*, 305(5937), pp. 827–829.

Tomii, K. and Akiyama, Y. (2004) ‘FORTE: a profile–profile comparison tool for protein fold recognition’, *Bioinformatics*, 20(4), pp. 594–595.

Trachana, K. *et al.* (2011) ‘Orthology prediction methods: a quality assessment using curated protein families’, *BioEssays*, 33(10), pp. 769–780.

Train, C.-M. *et al.* (2017) ‘Orthologous Matrix (OMA) algorithm 2.0: more robust to asymmetric evolutionary rates and more scalable hierarchical orthologous group inference’, *Bioinformatics*, 33(14), pp. i75–i82.

Treangen, T. J. and Salzberg, S. L. (2011) ‘Repetitive DNA and next-generation sequencing: computational challenges and solutions’, *Nature Reviews Genetics*, 13(1), pp. 36–46.

Tsai, K.-W. *et al.* (2015) ‘Evaluation and application of the strand-specific protocol for next-generation sequencing’, *BioMed Research International*, 2015, p. 182389.

Twyford, A. D. and Ennos, R. A. (2012) ‘Next-generation hybridization and introgression’, *Heredity*, 108(3), pp. 179–189.

Unver, T. *et al.* (2017) ‘Genome of wild olive and the evolution of oil biosynthesis’, *Proceedings of the National Academy of Sciences of the United States of America*, 114(44), pp. E9413–E9422.

URGI (2009) *URGI platform*. Available at: https://urgi.versailles.inra.fr.

Ureta-Vidal, A., Ettwiller, L. and Birney, E. (2003) ‘Comparative genomics: genome-wide analysis in metazoan eukaryotes’, *Nature Reviews Genetics*, 4(4), pp. 251–262.

Usadel, B. *et al.* (2009) ‘Co-expression tools for plant biology: opportunities for hypothesis generation and caveats’, *Plant, Cell & Environment*, 32(12), pp. 1633–1651.

Usadel lab—Jülich Research Centre/RWTH Aachen University (2014-2019) *plaBiPD.* Available at: https://www.plabipd.de (Accessed: 2 December 2019).

Utturkar, S. M. *et al.* (2014) 'Evaluation and validation of de novo and hybrid assembly techniques to derive high-quality genome sequences', *Bioinformatics*, 30(19), pp. 2709–2716.

Van Bel, M. et al. (2012) 'Dissecting plant genomes with the PLAZA comparative genomics platform', *Plant physiology*, 158(2), pp. 590–600.

Van Verk, M. C. *et al.* (2013) 'RNA-Seq: revelation of the messengers', *Trends in Plant Science*, 18(4), pp. 175–179.

Vezzi, F., Narzisi, G. and Mishra, B. (2012) 'Feature-by-feature--evaluating de novo sequence assembly', *PLOS ONE*. Public Library of Science, 7(2), p. e31002.

Vilella, A. J. *et al.* (2009) 'EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates', *Genome Research*, 19(2), pp. 327–335.

Vogt, G., Etzold, T. and Argos, P. (1995) 'An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited', *Journal of Molecular Biology*, 249(4), pp. 816–831.

Vollger, M. R. *et al.* (2019) 'Long-read sequence and assembly of segmental duplications', *Nature Methods*, 16(1), pp. 88–94.

Voshall, A. and Moriyama, E. N. (2018) 'Next-generation transcriptome assembly: strategies and performance analysis', in Abdurakhmonov, I. Y. (ed.) *Bioinformatics in the Era of Post Genomics and Big Data*. London: IntechOpen, pp. 15–36.

Voshall, A. and Moriyama, E. N. (2019) 'Next-generation transcriptome assembly and analysis: Impact of ploidy', *Methods*.

Vukašinović, N. *et al.* (2014) 'Dissecting a Hidden Gene Duplication: The

Arabidopsis thaliana SEC10 Locus', *PLOS ONE*, 9(4), p. e94077.

Wade, J. T. and Grainger, D. C. (2014) 'Pervasive transcription: illuminating the dark matter of bacterial transcriptomes', *Nature Reviews Microbiology*, 12(9), pp. 647–653.

Wain, H. M. *et al.* (2002) 'Guidelines for human gene nomenclature', *Genomics*, 79(4), pp. 464–470.

Wallace, I. M. *et al.* (2006) 'M-Coffee: combining multiple sequence alignment methods with T-Coffee', *Nucleic Acids Research*, 34(6), pp. 1692–1699.

Wang, B. *et al.* (2016) 'Unveiling the complexity of the maize transcriptome by single-molecule long-read sequencing', *Nature Communications*, 7, p. 11708.

Wang, B. *et al.* (2019) 'Reviving the Transcriptome Studies: An Insight Into the Emergence of Single-Molecule Transcriptome Sequencing', *Frontiers in Genetics*, 10, p. 384.

Wang, E. T. *et al.* (2008) 'Alternative isoform regulation in human tissue transcriptomes', *Nature*, 456(7221), pp. 470–476.

Wang, S. and Gribskov, M. (2017) 'Comprehensive evaluation of de novo transcriptome assembly programs and their effects on differential gene expression analysis', *Bioinformatics*, 33(3), pp. 327–333.

Wang, W. *et al.* (2014) 'Cassava genome from a wild ancestor to cultivated varieties', *Nature Communications*, 5, p. 5110.

Wang, Z., Chen, Y. and Li, Y. (2004) 'A brief review of computational gene prediction methods', *Genomics, Proteomics & Bioinformatics*, 2(4), pp. 216–221.

Wang, Z., Gerstein, M. and Snyder, M. (2009) 'RNA-Seq: a revolutionary tool for transcriptomics', *Nature Reviews Genetics*, 10(1), pp. 57–63.

Wan, X.-F. and Xu, D. (2005) ‘Computational methods for remote homolog identification’, *Current Protein & Peptide Science*, 6(6), pp. 527–546.

Waterhouse, A. M. *et al.* (2009) ‘Jalview Version 2—a multiple sequence alignment editor and analysis workbench’, *Bioinformatics*, 25(9), pp. 1189–1191.

Weber, E. H. (1834) *De pulsu, resorptione, auditu et tactu: Annotationes anatomicae et physiologicae*. Leipzig: Koehler.

Wee, Y. *et al.* (2019) ‘The bioinformatics tools for the genome assembly and analysis based on third-generation sequencing’, *Briefings in Functional Genomics*, 18(1), pp. 1–12.

Weibull, W. (1951) ‘A statistical distribution function of wide applicability’, *Journal of Applied Mechanics*, 18(3), pp. 293–297.

Wei, C. and Brent, M. R. (2006) ‘Using ESTs to improve the accuracy of de novo gene prediction’, *BMC Bioinformatics*, 7, p. 327.

Wei, L. *et al.* (2002) ‘Comparative genomics approaches to study organism similarities and differences’, *Journal of Biomedical Informatics*, 35(2), pp. 142–150.

Weston, J. *et al.* (2004) ‘Protein ranking: from local to global structure in the protein similarity network’, *Proceedings of the National Academy of Sciences of the United States of America*, 101(17), pp. 6559–6563.

Whelan, S. and Goldman, N. (2001) ‘A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach’, *Molecular Biology and Evolution*, 18(5), pp. 691–699.

Whelan, S., Liò, P. and Goldman, N. (2001) ‘Molecular phylogenetics: state-of-the-art methods for looking into the past’, *Trends in Genetics*, 17(5), pp. 262–272.

Wilks, S. S. (1938) ‘The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses’, *Annals of Mathematical Statistics*, 9(1), pp.

60–62.

Wittwer, L. D. *et al.* (2014) ‘Speeding up all-against-all protein comparisons while maintaining sensitivity by considering subsequence-level homology’, *PeerJ*, 2, p. E607.

Wolf, J. B. W., Lindell, J. and Backström, N. (2010) ‘Speciation genetics: current status and evolving approaches’, *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 365(1547), pp. 1717–1733.

Wright, A. F. (2003) in Cooper, D. (ed.) *Encyclopedia of the Human Genome*. London: Nature Publishing Group, pp. 959–968.

Xia, E. *et al.* (2019) ‘The tea plant reference genome and improved gene annotation using long-read and paired-end sequencing data’, *Scientific Data*, 6(1), p. 122.

Xie, Y. *et al.* (2014) ‘SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads’, *Bioinformatics*, 30(12), pp. 1660–1666.

Yandell, M. and Ence, D. (2012) ‘A beginner’s guide to eukaryotic genome annotation’, *Nature Reviews Genetics*, 13(5), pp. 329–342.

Yang, Y. *et al.* (2011) ‘Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates’, *Bioinformatics*, 27(15), pp. 2076–2082.

Yang, Z. (1994) ‘Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods’, *Journal of Molecular Evolution*, 39(3), pp. 306–314.

Yang, Z. (1996) ‘Among-site rate variation and its impact on phylogenetic analyses’, *Trends in Ecology & Evolution*, 11(9), pp. 367–372.

Yang, Z. (2014) *Molecular Evolution: A Statistical Approach*. Oxford: Oxford University Press.

Yang, Z. and Rannala, B. (2012) ‘Molecular phylogenetics: principles and practice’, *Nature Reviews Genetics*, 13(5), pp. 303–314.

Yoon, B.-J. (2009) ‘Hidden Markov Models and their Applications in Biological Sequence Analysis’, *Current Genomics*, 10(6), pp. 402–415.

You, Q. *et al.* (2018) ‘Development and Applications of a High Throughput Genotyping Tool for Polyploid Crops: Single Nucleotide Polymorphism (SNP) Array’, *Frontiers in Plant Science*, 9, p. 104.

Yu, D. *et al.* (2013) ‘Review of biological network data and its applications’, *Genomics & Informatics*, 11(4), pp. 200–210.

Zainuddin, I. M. *et al.* (2012) ‘Robust transformation procedure for the production of transgenic farmer-preferred cassava landraces’, *Plant Methods*, 8(1), p. 24.

Zerbino, D. R. *et al.* (2018) ‘Ensembl 2018’, *Nucleic Acids Research*, 46(Database issue), pp. D754–D761.

Zerbino, D. R. and Birney, E. (2008) ‘Velvet: algorithms for de novo short read assembly using de Bruijn graphs’, *Genome Research*, 18(5), pp. 821–829.

Zhang, M. Q. (2002) ‘Computational prediction of eukaryotic protein-coding genes’, *Nature Reviews Genetics*, 3(9), pp. 698–709.

Zhang, Q. and Backström, N. (2014) ‘Assembly errors cause false tandem duplicate regions in the chicken (Gallus gallus) genome sequence’, *Chromosoma*, 123(1-2), pp. 165–168.

Zhang, R. *et al.* (2017) ‘A high quality Arabidopsis transcriptome for accurate transcript-level analysis of alternative splicing’, *Nucleic Acids Research*, 45(9), pp. 5061–5073.

Zhang, S. V., Zhuo, L. and Hahn, M. W. (2016) ‘AGOUTI: improving genome assembly and annotation using transcriptome data’, *GigaScience*, 5(1), p. 31.

Zhang, X. *et al.* (2015) ‘Polyribosomal RNA-Seq reveals the decreased complexity and diversity of the Arabidopsis translatome’, *PLOS ONE*, 10(2), p. e0117699.

Zhao, Q.-Y. *et al.* (2011) ‘Optimizing de novo transcriptome assembly from short-read RNA-Seq data: a comparative study’, *BMC Bioinformatics*, 12 Suppl 14, p. S2.

Zheng, G. X. Y. *et al.* (2016) ‘Haplotyping germline and cancer genomes with high-throughput linked-read sequencing’, *Nature Biotechnology*, 34(3), pp. 303–311.

Zhou, S. *et al.* (2002) ‘A whole-genome shotgun optical map of Yersinia pestis strain KIM’, *Applied and Environmental Microbiology*, 68(12), pp. 6321–6331.

Zhu, B.-H. *et al.* (2016) ‘PEP_scaffolder: using (homologous) proteins to scaffold genomes’, *Bioinformatics*, 32(20), pp. 3193–3195.

Zhu, X. *et al.* (2015) ‘misFinder: identify mis-assemblies in an unbiased manner using reference and paired-end reads’, *BMC Bioinformatics*, 16, p. 386.

Zielezinski, A. *et al.* (2017) ‘Alignment-free sequence comparison: benefits, applications, and tools’, *Genome Biology*, 18(1), p. 186.

Zielezinski, A. *et al.* (2019) ‘Benchmarking of alignment-free sequence comparison methods’, *Genome Biology*, 20(1), p. 144.

Zimin, A. V. *et al.* (2017) ‘The first near-complete assembly of the hexaploid bread wheat genome, Triticum aestivum’, *GigaScience*, 6(11), pp. 1–7.

Zuckerkandl, E. and Pauling, L. (1965) ‘Evolutionary Divergence and Convergence in Proteins’, in Bryson, V. and Vogel, H. J. (eds) *Evolving Genes and Proteins*. New York: Academic Press, pp. 97–166.

Zynda, G. (2014) *Exponential Growth of NCBI Genomes*. Available at: http://gregoryzynda.com/ncbi/genome/python/2014/03/31/ncbi-genome.html

(Accessed: 8 November 2017).