Establishing graph based pan-genomics in *Arabidopsis thaliana*

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von
M.Sc. Christian Kubica
aus Oberhausen

Tübingen
2024

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	12.11.2024
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Detlef Weigel
2. Berichterstatter:	Prof. Dr. Daniel Huson

Abstract

By definition, single reference genomes cannot reflect genetic diversity. The representation of the genetic potential of a whole species as a single linear string of characters and all analyses based on them are inherently biased. This reference bias has been acknowledged for a long time, but only recently have we been able to address it. The advent of long-read sequencing and many additional genome assemblies for the same species has allowed us to obtain a better understanding of variation in genome content within a species. In addition, the availability of these new data types have made the implementation of a long standing concept feasible: the genome graph. This data structure combines multiple reference genomes into a single representation that is able to reflect more of the sequence space than a linear reference genome.

In this thesis I present six highly contiguous *de-novo* assembled genomes of *Arabidopsis thaliana* that are annotated using the new pan-genome aware *auto-ant* annotation pipeline. These assemblies are used to construct a complex, whole-genome alignment derived genome graph. I will show that building such a graph is not only theoretically possible, but also practically feasible, representing the full pan-genome of the input genome assemblies. I can access this graph-based pan-genome using the novel reference free variant detection algorithm *panSV*. I can also show that short-read alignments to the genome graph are possible and suffer from a reduced reference bias, due to the expanded reference structure. Variant calls based on the graph have a reduced heterozygosity noise that will aid future discoveries.

The use of genome graphs greatly increases our understanding of a species pan-genome and allows us to combine the power of multiple assembled genomes. Although the method is in need of further development and improvements, I have made a first case for the use of highly complex graphs in plant species.

Kurzfassung

Per Definition können einzelne Referenzgenome keine genetische Varianz darstellen. Die Darstellung des genetischen Potenzials einer ganzen Spezies als eine einzige lineare Kette von Merkmalen und alle darauf basierenden Analysen sind von Natur aus verzerrt. Diese Verzerrung durch lineare Referenzgenome ist seit langem bekannt, aber erst seit kurzem sind wir in der Lage, etwas dagegen zu unternehmen. Das Aufkommen der Long-Read-Sequenzierung und vieler zusätzlicher Genome für ein und dieselbe Spezies hat es uns ermöglicht, ein besseres Verständnis für das wahre genetische Potenzial einer Spezies zu gewinnen. Darüber hinaus hat die Verfügbarkeit dieser Genome die Umsetzung eines seit langem bestehenden Konzepts möglich gemacht: den Genomgraphen. Diese Datenstruktur kombiniert mehrere Referenzgenome zu einer gemeinsamen Darstellung, die mehr vom Sequenzraum abdecken kann als ein einzelnes lineares Referenzgenom.
In dieser Arbeit stelle ich sechs *de-novo* assemblierte Genome von *Arabidopsis thaliana* vor, die mit einer neuen annotations Pipeline, *auto-ant*, annotiert wurden. Diese Genome werden verwendet, um einen komplexen, auf einem whole-genome Alignment basierenden Graphen zu generieren. Ich werde zeigen, dass die Erstellung eines solchen Graphen nicht nur theoretisch möglich, sondern auch praktisch durchführbar ist, und er in der Lage ist, das Pangenom der genutzten Genome zu repräsentieren. Auf die variablen Regionen dieses Pan-Genoms kann ich mit dem neuartigen referenzfreien Algorithmus *panSV* zugreifen. Ich kann auch zeigen, dass short-read Alignments gegen den Genomgraphen möglich sind und diese aufgrund der erweiterten Referenzstruktur eine geringere Heterozygotenrate aufweisen.
Die Verwendung von Genomgraphen kann unser Verständnis der genetischen Diversität einer Spezies erheblich verbessern. Obwohl diese Methode noch weiterentwickelt und verbessert werden muss, habe ich ein erstes Beispiel für die Verwendung hochkomplexer Graphen bei einer Pflanzenart geliefert.

Contents

1 Introduction **1**
- 1.1 Reference genomes . . . 2
  - 1.1.1 History of reference genomes . . . 3
  - 1.1.2 Utilization of reference genomes . . . 4
  - 1.1.3 Reference bias . . . 8
- 1.2 Genome graphs . . . 9
  - 1.2.1 Types of graphs . . . 10
  - 1.2.2 Graph construction . . . 11
  - 1.2.3 Alignments to graphs . . . 13
  - 1.2.4 Usability of graphs . . . 14
- 1.3 Pan-genomics . . . 15

2 Methods **17**
- 2.1 Computational tools . . . 17
  - 2.1.1 *auto-ant* . . . 17
  - 2.1.2 *panSV* . . . 23
- 2.2 The sixRef project . . . 27
  - 2.2.1 Assembly . . . 27
  - 2.2.2 SV calling & evaluation . . . 28
  - 2.2.3 Annotation . . . 30
  - 2.2.4 Graph construction & processing . . . 31
  - 2.2.5 Graph alignment evaluation . . . 34
  - 2.2.6 Graph genotyping . . . 36

3 Results **41**
- 3.1 Generation & annotation of new genome assemblies . . . 41
  - 3.1.1 Genome assemblies . . . 41
  - 3.1.2 SV calling and comparison to the *TAIR10* reference genome . . . 43
  - 3.1.3 SV set comparison . . . 47
  - 3.1.4 Genome annotation . . . 51
- 3.2 Graph genome . . . 61
  - 3.2.1 Graph construction . . . 61
  - 3.2.2 Graph pan-genome . . . 62
  - 3.2.3 Graph SV calling . . . 65
  - 3.2.4 Graph alignment evaluation . . . 70

3.2.5 Graph genotyping 72

4 Discussion **83**
4.1 The sixRef pan-genome 84
4.2 The sixRef proteome 86
4.3 Graph genome 87
4.4 Graph based short-read alignments 89

5 Conclusion & Outlook **95**

A Abbreviations & Glossary **99**

B Supplementary **101**
B.1 Supplementary Figures 101
B.2 Supplementary Tables 106

Bibliography **113**

Chapter 1

Introduction

In the "Hitchhiker's Guide to the Galaxy" by Douglas Adams (Adams, 1995) a giant super computer called Deep Thought calculates the answer to "the Ultimate Question of Life, the Universe, and Everything" - and comes with a simple result: 42. When the characters tasked with obtaining the answer from the computer vent their frustration at its lack of usefulness, the computer points out that the question was hardly well formulated. A colleague recently compared Deep Thought to genome graphs, the central element of this thesis - a near-mythical construct capable of answering almost all important questions of population genetics. In other words, an incredibly powerful tool if only we can formulate the right questions to ask. In this thesis, I will explore methods to construct genome graphs and describe their usability in the model organism *Arabidopsis thaliana*. I will present methods to describe the pan-genome stored in a graph built from six *de-novo* assembled accessions, and access it to describe the variation in the larger population of accessions in the 1001 Genomes Projects, for which short-reads were available (1001 Genomes Consortium, 2016).

The genome of an individual is a mosaic of genetic information inherited from its ancestors and it contains the instructions not only for building the cells and organs of the individual, but also how it functions in response to the environment. Importantly, genetic information is not static, and spontaneous mutations continuously fuel genome diversification, on which in turn selection acts to favor those variants that are particularly advantageous. The extent of genetic variation in species such as ours cannot be overestimated. The challenge has been to capture and describe genetic diversity. A good starting point are the so-called reference genomes. Reference genomes are near-complete representations of the genetic material of a species as strings of the four nucleotides G, A, T, and C. They are the coordinate system to locate genetic features such as genes, or microRNAs, and the frame that sequence variation is compared to. Reference genomes are obtained by sequencing the DNA of either a single individual or a mix of individuals and assembling the sequencing fragments into a linear string of bases using computational methods. Typically, even though many organisms are diploid, the reference genome is haploid. By its nature, a linear reference genome is unable to represent the genetic diversity of a species that consists of multiple individuals, and thus any analysis it is used for will be intrinsically biased towards the variant combination represented in the reference

genome. Advances in sequencing technologies and computing power are giving rise to a better option: Multiple reference genomes can be combined into one single reference structure, where the shared sequence is compressed, but the diverged sequence remains represented and therefore available for interrogation. This structure is known as a genome graph and enables us to better represent the genetic potential of a population. We can now have access to a variety of different alleles in order to better describe the true cause of phenotypic differences.

1.1 Reference genomes

Even though genetic diversity is obvious and has long been appreciated, the first published genomes were all advertised as "the genome of species XYZ" (Venter *et al.*, 2001). Already when the project to sequence a human genome was conceived, Bruce Walsh and Jon Marks wrote a letter to Nature stating their concerns (Walsh and Marks, 1986), that a linear reference genome can never truly represent the genetic sequence of a diploid organism, let alone of a population. Despite its shortcomings, single reference genomes have been used successfully to describe the genetic diversity of populations (Dujon, 1996; 1000 Genomes Project Consortium *et al.*, 2010; 1001 Genomes Consortium, 2016). Reference genomes have been used to predict, define and anchor functional elements, such as genes. More importantly, they establish a framework for comparison with other individuals. By resequencing parts of an individual's genome or the full genome and comparing the sequencing reads to this established framework, the reference can be interrogated for genetic differences between the two individuals. While resequencing projects have been successful in describing much of the small-scale variation of a population, they also highlighted one of the major problems of single, linear reference genomes. It is easy to represent and detect sequences that are absent from the query genome, but challenging to describe novel sequences. This introduces a reference bias (Degner *et al.*, 2009), which will be discussed in more detail in subsection 1.1.3. Although this is common knowledge, at the time that I started the dissertation work, relatively little progress had been made in solving this problem.
Since the construction and publication of the first reference genomes, the quality and quantity of available references has seen an exponential increase. Genomic resources for popular model species are no longer based on a single individual, but try to include diverse sequences from multiple donors (Shukla *et al.*, 2019; Dewey *et al.*, 2011), while still maintaining a linear structure. The latest release of the human genome even includes multiple versions of highly diverged regions (Schneider *et al.*, 2016), which can be utilized for read mappings by the common read mapper bwa (Li, 2013). A drop in sequencing cost has also made it possible to assemble more and more reference genomes for uncommon species (Al-Mssallem *et al.*, 2013; Stevens *et al.*, 2016; Nowoshilow *et al.*, 2018), and with the advent of cheap long-read sequencing, multiple genome assemblies of similar quality are available for a single species (Zapata *et al.*, 2016; Jiao and Schnee-

berger, 2020). This has extended our knowledge of structural variation, enabling us to detect more and larger variants, something that had previously been limited by read length and assembly quality. Thus, a new field in genomics emerged: pan-genomics, where scientists try to describe as much of the shared and variable sequence of a species as possible.

1.1.1 History of reference genomes

The assembly of nucleotide sequence fragments into longer sequences goes back many decades (Sanger and Thompson, 1953; Min Jou *et al.*, 1972), but the first complete genome sequenced was that of the 3,569 bp long RNA sequence of the bacteriophage *MS2* (Fiers *et al.*, 1976), followed by genomes of other viruses and of bacteria (Baer *et al.*, 1984; Sodeik *et al.*, 1993; Blattner *et al.*, 1997). The first eukaryote, a strain of baker's yeast, *Saccharomyces cerevisiae*, with a combined assembly size of 12 Mb, saw its genome sequence completed in 1996 in an combined effort of 94 laboratories (Goffeau *et al.*, 1996), followed by genomes of the first multicellular organism, *Caenorhabditis elegans* (97 Mb), in 1998 (C. elegans Sequencing Consortium, 1998). Thereafter more and more complex genomes were assembled in close succession. In March 2000 the first genome for *Drosophila melanogaster* (120 Mb) (Adams *et al.*, 2000), and in December of the same year the first genome for *A. thaliana* (115 Mb) (Arabidopsis Genome Initiative, 2000) were released. The first complete human genome was published in February 2001 (Venter *et al.*, 2001). Since then assembled genomes from many more organisms have been added, and even more individuals or strains have been resequenced.
Sequencing a genome is rarely a straightforward endeavor. In an ideal case one would be able to read the entire molecule of each chromosome with high accuracy. In reality we are rarely able to even isolate a full chromosome for end-to-end sequencing, but have to painstakingly reconstruct the genome from overlapping sequence fragments. The length and accuracy of these fragments defines the ability to resolve complex regions in the genome. For example, repeats that are longer than the fragment length will not be resolved to their precise copy number, and duplicated sequences can create ambiguity in the assembly. An additional factor that complicates genome assemblies is the genome size. Larger genomes not only contain more genes, but also more repeats and other regions that are challenging to assemble. As a result, assemblies often consist of continuous fragments, or contigs, that have been built from uniquely overlapping sequence fragments. These contigs can then be ordered and oriented using additional information, such as optical maps, or chromatin contact maps. While complete sequences for the relatively small circular genomes of bacteria, which has also few repeats, have been available for years, the first telomere-to-telomere assemblies of multicellular eukaryotes have only recently begun to emerge (Wang *et al.*, 2021; Gonzalez de la Rosa *et al.*, 2021; Giguere *et al.*, 2021; Brázda *et al.*, 2022; Chen *et al.*, 2023) and most genome assemblies currently in use contain gaps that could not be bridged by sequencing technologies. Since the invention of sequencing methods, the length of obtainable sequence reads has gradually

increased. Increased read length often came with an increase in cost and loss of accuracy. Therefore the choice of sequencing technology, and assembly method is always a trade off between cost, accuracy, and speed to obtain the best result for the question at hand. While prices for new sequencing technologies were increasing initially, cost for traditional technologies decreased. An example for the drastic drop in the cost of genome assemblies comes from *D. melanogaster*. While the first genome assembly, published in 2000, was a complex endeavor, technically and financially, the cost of sequencing a *D. melanogaster* genome had dropped from an estimated 1.8 billion dollar (Kris A. Wetterstrand, 2019) to 1,000 dollar in 2018 (Solares *et al.*, 2018), and was reported to be 350 $ per genome in 2020 (Kim *et al.*, 2020). In addition to dropping costs the assembly of longer and more complex genomes became possible, such as that of sugar cane, at 31 Gb (Stevens *et al.*, 2016), or the mexican walking fish, better known as axolotl, with an assembly of 32 Gb (Nowoshilow *et al.*, 2018). In contrast, the current reference assembly of *A. thaliana* (Berardini *et al.*, 2015), with a size of 119 Mb, is among the shorter genomes. Nevertheless the use of improved long read sequencing methods has led to gains in the known sequence space of the reference accession. The latest adding another 14.6 Mb to the known sequence (Wang *et al.*, 2021).

1.1.2 Utilization of reference genomes

While, in the first place, reference genomes provide a mere framework to anchor features and genetic differences, they have become an invaluable tool in genetics. If we are to make sense of the long string of bases (A,G,C and T), we need to annotate it, or compare it to another string. Different types of functional elements in a genome, such as genes, transposable elements (TEs), or repetitive regions, can be detected and annotated based on the reference sequence itself. Other features can only be detected and anchored with the help of additional, external, knowledge, such as epigenetic modifications, or by comparison with another sequence, such as sequence variation. These annotations help us to better understand the genetic potential, as well as the evolutionary forces that shape genomes. In the following subsections we will dive deeper into genome annotation and the detection of sequence variation from reference genomes.

Genome annotation

The detection of structural differences between two genomes holds little value on its own. Only in the light of functional annotations do they hold true power. Therefore one of the uses of an assembly is to serve as a template for the annotation of functional elements like genes and TEs. Over time, different annotation methods have been developed and refined. The two main approaches for sequence annotation are either *ab-initio* or *evidence based* predictions. *Ab-initio* predictions rely on statistical models to detect similarities to known motifs in the sequence. Algorithms like *SNAP* (genes) (Korf, 2004), *repeatMasker* (repetitive regions) (Smit, AFA, Hubley, R & Green, P., 2013), or *EDTA*

(TEs) (Ou *et al.*, 2019) scour the genome for predetermined sequence motifs. Machine learning has also found its way into the field of genome annotation and is for example used by *DeepAnnotator* (Amin *et al.*, 2018). Such motifs, or models are based on previously identified high confidence features and as such these methods are unable to identify features that have not been observed before, but come with a lower price tag as no additional data generation is required. In contrast, *evidence based* annotation approaches utilize external sequence information, such as transcription data, or gene sequences that are aligned to the reference sequence. A very straightforward approach is applied by tools like *exonerate* (Slater and Birney, 2005), or *LiftOff* (Shumate and Salzberg, 2020), where an existing annotation is lifted to the new assembly based on sequence alignments. Despite its simplicity it suffers from the very same bias that our reference genomes suffer from. It is only possible to annotate features that are already known. In contrast *Cufflinks* (Trapnell *et al.*, 2010) relies on RNA sequencing data and assembles the RNA evidence into transcripts. While in theory this method has the best biological evidence supporting their results, it relies on the expression of the given genes, which is not alway the case and can vary based on the tissue of origin and time of collection. To improve the results, and overcome their individual weaknesses, methods that combine these approaches have been developed, such as *augustus* (Stanke *et al.*, 2008). Despite being classified as an *ab-initio* annotation tool, it can utilize and combine layers of extrinsic evidence to either support the prediction or revoke it.

Genetic variation

Genetic variation shapes inter- and intra-specific phenotypic diversity and can have major implications for the organism. This variation occurs on different scales, ranging from single base point mutations to alterations of entire chromosome arms. Variation also takes different forms. A sequence can be substituted, moved to a different position, copied, lost, or gained. The underlying mechanisms are diverse and in some cases remain unknown. Scientists distinguish between two groups of variation: small variation, such as single nucleotide polymorphisms (SNPs) and small insertion-deletion events (InDels), and large structural variation (SV) with a size greater than 50 bp, that can alter the structure of the entire genome. Due to constraints in sequencing technology only a fraction of this variation could be studied with sufficient confidence as the size and type of detectable variation has been severely limited (1001 Genomes Consortium, 2016; Caicedo *et al.*, 2007; Sudmant *et al.*, 2015; Durvasula *et al.*, 2017; Fulgione *et al.*, 2018). Especially the identification of structural variation has been impeded by technological boundaries. Such structural variations appear in many forms, and due to their larger size, can contain additional nested variation, which makes their identification, categorization, and description even more challenging. The easiest form of structural variation is the binary presence or absence of sequence from one of the genomes. The commonly used names, insertions and deletions imply a polarity, based on the reference genome. This polarity is not biologically backed, therefore the phrase presence absence variation (PAV) is becom-

ing more popular. While other structural variants can also seem like PAV-type variation at first glance they are changes in copy number or sequence localisation. In case of copy number changes additional copies of a sequence, e.g. a TE, are gained or lost in one of the individuals and can then evolve independently, resulting in nested, smaller variants that can help to distinguish multiple copies of the same sequence. Rearrangements, without copy number changes can also happen, resulting in local presence-absence type events. A more obscure and not that obvious structural variant is a change in ploidy. This does not leave directly detectable traces in the genome but in the smaller variants due to an independent evolution. Only recently have studies begun to unravel the impact of structural variation on a larger scale (Jiao and Schneeberger, 2020).
While the impact of point mutations has been described extensively over the last decades, their impact is limited by their size. A simple base pair change can result in a change of a single amino acid, a shift in reading frame, or a premature stop codon, that affect only a single gene. In contrast, due to the sheer number of bases affected, large structural variants regularly cover whole gene clusters and have been reported to have a profound impact on the individual, such as disease traits (Beyter *et al.*, 2020). As such the selective pressure on large SVs is stronger than on small variants, and thus only a fraction of SVs become fixed in a population. This means that SVs, fixed in the population, tend to have a functional role. The analysis of such rare SVs can shed light into the underlying mutational processes (Abel *et al.*, 2020). They have been shown to be a driving factor in agricultural plant breeding (Walkowiak *et al.*, 2020; Jayakodi *et al.*, 2020). The role of inversions in particular has been studied extensively in plant breeding and adaptation. Inversions have been shown to play a role in adaptation to salt (Lowry and Willis, 2010) and changes in flowering time (Fransz *et al.*, 2016; Göktay *et al.*, 2020). One inversion in barley is suspected to be the causal variant for its adaptation to the climate of the western hemisphere (Jayakodi *et al.*, 2020). They have also been linked to changes in grain size of basmati rice (Choi *et al.*, 2020), grape color (Zhou *et al.*, 2019) and the domestication of tomato (Alonge *et al.*, 2020; Soyk *et al.*, 2019). Another major contributor is the mobilome, which in plants is largely composed of TEs. TEs arise with a frequency comparable to SNPs, but have a bigger impact, due to their size (Baduel *et al.*, 2021). Such insertions can easily have a functional impact and become fixed in populations (Walkowiak *et al.*, 2020; Hufford *et al.*, 2021). They have been shown to impact gene expression in *A. thaliana* and maize (Hollister *et al.*, 2011; Noshay *et al.*, 2020), to contribute to the domestication of maize (Studer *et al.*, 2011), and the sub-genome speciation of rice (Ma *et al.*, 2020). Such rapid changes in the genome can become advantageous for rapid adaptation to new environments (Xu *et al.*, 2009; Van de Weyer *et al.*, 2019). Even copy number variation in tandem repeats have been shown to have phenotypic impact. As these types of variant have been hard to unroll in linear, incomplete references their impact on body height, hair morphology and human health biomarkers has only recently been discovered (Mukamel *et al.*, 2021). The duplication of whole genes leaves one copy open for mutational processes, letting them accumulate additional mutations. This affects mainly species specific and non-essential genes (Zmienko *et al.*, 2020).

The detection of such differences between the reference and a second individual depends on our ability to represent the variation with the data at hand and to interpret the result. In the past two general approaches, with different advantages and disadvantages, have been used to detect and classify the genetic changes between the reference genome and another individual. Either, relatively inexpensive, but fragmented, whole genome shotgun sequencing, or a, more expensive and time consuming, whole genome assembly was constructed. In both cases the resulting sequences were then aligned and compared to the existing reference genome and differences were described in the context of the reference genome. In the most common case of fragmented short-read alignments, this process is heavily biased by the length of the fragments and the quality of the reference genome. Each fragment needs to be uniquely placed in order to describe the sequence differences. For variants shorter than the average fragment length this difference is easy to interpret. Here the bases that do not align are recorded. In general such methods rely on reads being aligned and a clear signal from multiple fragments covering the same position. Tools, like *FreeBayes* (Garrison and Marth, 2012), or *GATK* (Poplin *et al.*, 2018), have become more and more elaborate, in order to reduce the uncertainty of such calls by identifying sequencing artifacts, and performing local realignments to reduce noise. This method introduces a bias towards the detection of SNPs and sequence missing in the query, over novel sequence and complex structural variants (Sudmant *et al.*, 2015); and while the newly emerged long read sequencing allows the detection of large SVs from continuous reads, they are generally unsuited for SNP detection as their higher error rate and problems with homopolymer resolution make calls less reliable. Recent advances in technology, like circular consensus reads, have mitigated this problem by again trading read length for accuracy. Longer variants are less obvious to detect as they do not necessarily leave clear traces in the alignment. They are identified more indirectly by observing changes in the coverage (e.g. deletions, repeat expansions), distance between paired reads (e.g. translocations), changes in orientation (e.g. inversions), or loose, unaligned ends of reads (e.g. insertions). Therefore most of these events require specialized tools to be detected, and even then the accuracy can vary. Well known tools in this category are *pindel* (Ye *et al.*, 2009), *DELLY* (Rausch *et al.*, 2012), *BreakDancer* (Fan *et al.*, 2014), and *GRIDSS* (Cameron *et al.*, 2021) that each focus on the detection of alignment breakpoints to localize the start and end positions of variable regions. Increased read length allows for the detection of longer variants, as they are more likely to bridge complete variants. Complete genome assemblies can be seen as ultra long reads that bridge most variants and, as such, circumvent some of these problems. The challenge here is rooted in the previously described incompleteness of the assembly, that itself might be based on short-reads and have incorrectly resolved regions. Some of these problems are mitigated by the use of long read sequencing technologies for the generation of the assembly. *Assemblytics* (Nattestad and Schatz, 2016) is one of the earlier tools to compare genome assemblies, while *SyRI* (Goel *et al.*, 2019) is a more recent development that puts more effort on the categorization of variants. Despite recent developments one of the main underlying problems persists. As the mechanisms that give rise to the forma-

tion of SVs are not well understood, the problem itself is not well defined. This means that tools that are looking to detect structural variants are biased towards a computational solution that does not necessarily reflect the underlying biology (Morrison, 2018).
The predefined vcf file format is most commonly used to store variants detected from pairwise sequence comparisons (Danecek *et al.*, 2011). It is a reference based format that represents variation in the context and coordinate system of a single reference genome. Variation is stored as reference and non-reference alleles and are attributed to the samples they were detected in. Thus it is a lossy format that is unable to represent the coordinate of a variant in multiple samples, in addition by its reference based nature it is unable to represent nested variation. Nested variation is a type of variant that is found within another larger variant. It mostly occurs when multiple pairwise comparisons are combined in a single vcf file. In the context of a vcf file such a variant would be represented as two large non-reference alleles that are almost identical except for the few bases of the nested variant. This by itself is a hindrance in the work on structural variants in large populations and pan-genomics that needs to be addressed in the near future.

1.1.3 Reference bias

As previously mentioned, traditional references only represent a single individual of the species studied - in fact just a single haplotype - as a linear string of bases. It will therefore never be able to fully integrate the genetic variety of a population or species and any analysis performed on this reference genome will always be biased towards this haplotype. When sequencing data from an individual containing an additional sequence is compared to a reference lacking this sequence, the reads will be aligned with the next best region of the genome, or remain unaligned. Unaligned sequences are mostly lost for deeper analysis, whereas misplacement of reads can alter the results of later analysis by obscuring actual signals. We can distinguish between two main effects of reference bias. The first being missing variant calls due to large chunks of missing sequence. For example the reference and annotation of the maize genome only contains an estimated 63% - 74% of the total gene number (Hirsch *et al.*, 2014; Lu *et al.*, 2015; Hirsch *et al.*, 2016; Jin *et al.*, 2016). This results in a large part of the coding sequence of this species being inaccessible for analysis and skews every result towards the fraction being present. In wheat one of the most commonly used reference genomes is based on the Chinese spring ecotype, that has been collected around 1900 and is thus missing a lot of interesting agronomic traits (Bayer *et al.*, 2022). The second effect is an overestimation of heterozygosity. When copies of a repetitive region are absent from the reference, the sequencing reads originating from this position will be aligned to the copies present in the reference genome and result in heterozygous variant calls. In *A. thaliana*, a species with a mostly homozygous genome, due to selfing as the preferred mode of reproduction, 44% of the variant calls are registered as heterozygous (Jaegle *et al.*, 2021). This is a direct result of the incomplete sequence representation in the reference genome. The severity of a reference bias correlates with the available read length. Longer sequence fragments

are more likely to span the full SV and find unique anchors on both sides, while shorter reads have a higher chance to result in spurious alignments. In general this skews the detectable variants towards SNPs, short variants, and events of absent sequence, as they are easier to detect (De Coster and Van Broeckhoven, 2019; Ho *et al.*, 2020). Even the interpretation of variant calls can be biased by an incomplete reference, as such calls do not necessarily represent the underlying evolutionary context that gave rise to this variant (Alonge *et al.*, 2020).
Different ideas and approaches have been implemented to mitigate the problems of linear reference genomes. One is to use multiple references in parallel and try to always choose the most appropriate allele based on the available references, as implemented in *reference flow* (Chen *et al.*, 2021). This approach is especially powerful if artificial references are used to represent sub-populations, but it suffers from the absence of a unifying coordinate space. A different approach is to offer either unplaced bait sequences to draw away spurious mappings from interesting regions, or the introduction of additional major-alleles that have been anchored to the existing reference. By collecting them from different donors this can mitigate the effects of the reference bias (Dewey *et al.*, 2011; Pritt *et al.*, 2018; Shukla *et al.*, 2019; Grytten *et al.*, 2020). These methods are a first step towards reducing the reference bias. As a result of previous studies, we now have a better understanding of the modularity and variability that genomes exhibit, apart from simple presence, absence variation, tackled by these methods. A linear reference can not hope to capture this degree of complexity, but graphs are an obvious data structure to do so. Large rearrangements or the presence and absence of sequences can be represented naturally, as well as fine grain-resolved base pair variation (Novak *et al.*, 2017; Ameur, 2019; Ballouz *et al.*, 2019).

1.2 Genome graphs

Genome graphs are a way to include and represent variation in an accessible and natural data structure. In mathematics graphs are used to describe relationships between objects. In genome graphs these objects are sequence fragments that are stored in nodes. The relationship between those nodes, represented in the graph, are their order and orientation, and are represented by edges connecting the nodes. We can then introduce coloured paths that traverse the connections made by edges to represent longer sequences that are made up of multiple consecutive nodes, such as contigs, or full chromosomes in the graph (Figure 1.1).
Although the concept of graphs in genomics is not new, it has been hindered by computational restraints and availability of data. Constructing genome graphs and mapping reads to them requires enormous computational resources that have only become available in recent years. First, simple but usable, concepts of genome graphs for read mappings were around as early as 2009, but remained underused (Schneeberger *et al.*, 2009). Nevertheless, graph structures have been used in several other areas of genomics, including

genome assemblies. Assembly graphs are able to represent ambiguous results of the assembly process. Such an assembly graph is later cut and collapsed into a linear reference sequence, removing all ambiguity (Pevzner *et al.*, 2001; Myers, 2005; Bankevich *et al.*, 2021). Another application is the representation of gene model splice forms. The different splicing options of RNA transcripts within a single individual can be represented in a graph structure. Here the nodes represent full exons that are connected to represent different isoforms in a splice graph (Rogers *et al.*, 2012; LeGault and Dewey, 2013). Such a graph can then be used as a reference to map RNA sequencing reads to (Denti *et al.*, 2018; Kim *et al.*, 2019). This enables researchers explore the frequency of different splice variants and even discover new ones. Due to their limited size and complexity, splice graphs pose a far smaller computational challenge than full genome graphs. The graphs currently in use that are closest to real genome graphs are the haplotype graphs employed by *GATK HaplotypeCaller* (Poplin *et al.*, 2018) to calculate the haplotype likelihood of a given set of sequencing reads against the background of detected variation.

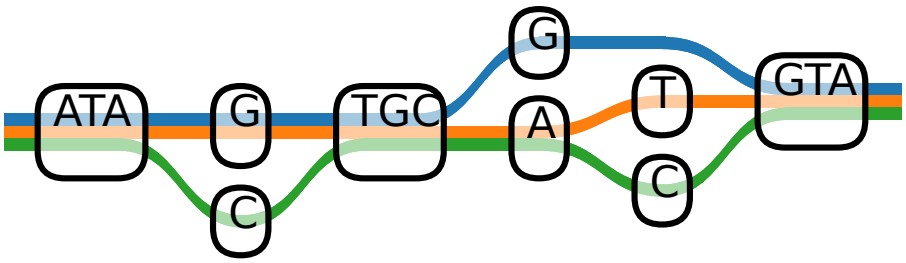

Figure 1.1: Genome graph - Example of a genome graph with three paths (blue, orange and green). Each path traverses multiple sequence-containing nodes. Some nodes are traversed by multiple paths, thus compressing the sequence space. This small graph contains multiple bubbles. One between the nodes ATA and TGC, and the second between the nodes A and GTA. In addition it contains a superbubble with nested variation, between TGC and GTA.

1.2.1 Types of graphs

Depending on their intended application, different types of graphs have been utilized in genomics, each with their own set of properties. A simple and widely used type of such graphs are k-mer based *de-Bruijn* graphs. In these graph structures, each node represents a single k-mer that is connected to all observed, overlapping k-1 mers in the set (Jackson *et al.*, 2010; Bankevich *et al.*, 2021). Compressed *de-Bruijn* graphs remove the overlapping redundancy of adjacent nodes and thus are shaped like classical genome graphs, with the only difference being that they are not built using an alignment. Due to their k-mer based construction *de-Bruijn* graphs disregard synteny information and are more compressed, and therefore more complex than alignment based graphs. A

completely different approach are cactus graphs that, by definition, enforce a linearity in their graph structure (Paten *et al.*, 2011). A cactus graph is always a directed, acyclic graph (DAG). Graphs are defined as directed, if their edges have directions, a feature that all genome graphs share. This is necessary in order to order and orientate the connected nodes. For simplicity some graphs are also acyclic, which means that a walk through such a graph can never return to a node it has visited before. Obviously this prevents the graph from compressing repetitive sequences and representing copy number variations and repeat expansions. Graphs that are able to represent the full spectrum of variation are based on whole genome alignments and can be very complex. Those graphs can be called 'connected graphs' due to their high number of connective edges. Although they come closest to a real representation of synteny and divergences of the real genomes, in some cases it will be necessary to simplify the graph structure to make it usable.
A genome graph represents alternative sequences and structural rearrangements in its topology. One of the most simple topology feature is a bubble. A bubble is defined by two anchoring nodes and consists of at least two different traversals to connect the two anchors. In any case a bubble is a closed structure that has no other way in, or out, except through its two anchors (Figure 1.1). The simplest biological example of a bubble is a SNP, where the sequence up- and downstream of it form the two anchors and the different bases are the traversals. A traversal in a graph describes a path through the graph that traverses edges and nodes in an order and orientation defined by the graph in order to represent a specific sequence. A traversal must consist of at least one node. This simple concept of a bubble can be nested in a way that a second, larger bubble contains a smaller bubble. These bubbles are called superbubbles (Onodera *et al.*, 2013; Dabbaghie *et al.*, 2021). This concept of classification holds true for cacti and DAGs, but falls short of representing the true structure of genomes. In order to do this we need to introduce the concept of snarls, that in contrast to bubbles do not require to be closed, but can have traversals touching just one anchor and leave the structure without touching the second anchor (Paten *et al.*, 2018).

1.2.2 Graph construction

Constructing a genome graph is not a trivial task and poses its own set of challenges. In order to build a graph we need to distinguish between shared and diverged sequences and combine them into a usable format. Shared sequence will be compressed into single nodes, whereas diverged regions open bifurcations in the graph structure. Depending on the question that is supposed to be answered with the graph multiple different approaches and input data types can be selected. Graphs can be built from pre-existing variation stored in vcf files, or alignments of whole genome assemblies and any type of sequencing data.
In principle a complete genome graph could be as simple as 5 nodes (A, G, C, T, N), each connected to the other ones and itself by edges. This graph can accurately represent an infinite amount of genomes. But its use would be very limited, because the detection of

subgraphs would be impossible and the coordinate system, imposed by syntenic regions would be completely lost. This problem illustrates the challenges that graph construction is facing: making sense of the complex genetic landscape, while keeping the graph usable and interpretable. One of the earliest approaches to graph construction have been *de-Bruijn* graphs. Their simple mathematical approach makes their construction very easy, but completely disregards the underlying forces that have shaped the genome. *De-Bruijn* graphs are build from overlapping k-mers. Depending on the k-mer size they contain a large number of nodes, and can be very complex and connected. In order to reduce the complexity of such graphs, while maintaining the simplicity of the graph construction the software *REVEAL* was developed (Linthorst *et al.*, 2015). It uses a k-mer chaining approach that identifies maximal unique matches (MUMs) from multiple input genomes and chains them. The approach is iteratively run in each local bubble of the chain. This results in a DAG as only MUMs are considered and no interaction between the sequence in different bubbles is possible. Similar graphs can be constructed by enriching a linear reference genome with variation imported from a vcf file. These are even more simple as they can not represent nested variation and due to the fact that it is being built from a vcf file means that the only coordinate system present is the one of the reference genome and the genetic context of the variants imported into the graph are lost. The graph software package *vg* (Garrison *et al.*, 2018) uses this method extensively and most of their tools are tailored around such graphs (Garrison *et al.*, 2018). While the graphs constructed using *minigraph* are similar to vcf infused graphs, they are built from alignments to a reference (Li *et al.*, 2020). Whole genomes, or long reads are aligned to a reference and regions that are different that surpass a defined distance threshold. They are added as nodes to the graph. This process can progressively be repeated to add variation from multiple individuals. As the added variation is defined by a genetic distance to the sequence that already exists in the graph, a large set of variants are omitted and information is lost. Furthermore the order of genomes in the progressive construction process influence the resulting graph. Similar to vcf based graphs, only the path of the reference genome persists and all other accessions are lost. Despite their simplicity such graphs are unable to capture the complex landscape of real genomes. This complexity can only be approached by fully aligning multiple sequences and retaining the paths defined by the input sequences. An intermediate approach is implemented in *progressiveCactus* (Armstrong *et al.*, 2020). The internal data structure is represented as a DAG, but it is able to retain all paths and make it available as a genome graph. The last method presented here is the most complex one and has only recently progressed to a usable form. It creates connected graphs and is based on a multiple whole genome alignment that is used to identify syntenic regions between the input genomes. Afterwards the alignment is converted into a graph structure and refined in multiple realignment steps in an effort to smooth the graph to retain synteny, but also represent the structural differences as accurately as possible. This approach has been implemented in the graph construction pipeline *pggb* (Garrison *et al.*, 2023). One of the main challenges of this approach is the creation of a multiple whole genome alignment. Although several tools for such align-

ments have been implemented, the underlying problem still remains unsolved (Li, 2018; Song *et al.*, 2022). Once a graph has been created additional variation can be added to it by aligning additional sequences to them. Variations stored in these alignments can then be augmented into the graph, adding more sequence and complexity. As new variation is added the graph topology and the number of nodes, and edges changes. This has massive implications for analysis that has been performed before and features that have been anchored to the graph. This makes updating existing graphs a tedious task.
As we are going to discuss next, graphs can not be indefinitely complex and still usable. For some analysis even simple DAGs can reduce the reference bias, while for others this will not suffice. This means the application of a genome graph needs to be well defined in order to choose the right construction method that balances the amount of sequence information needed with the complexity of the graph.

1.2.3 Alignments to graphs

Although the graph itself already holds valuable information, its true potential lies in its use as an alternative reference structure. The most basic of them being the target for sequence alignments. In the process of establishing genome graphs as alternative reference structures this is the first step to unlock its full potential. Thus far multiple graph alignment methods have been implemented. The earliest methods to map sequencing reads to graph structures were implemented in an attempt to augment variation into linear references to improve the accuracy and mapping score while maintaining the original reference system (Schneeberger *et al.*, 2009; Kim *et al.*, 2019). These graphs bear little resemblance with modern genome graphs in complexity and variant resolution. In contrast to the simple sequence space of linear references, or the reduced complexity of earlier attempts, the complex nature of genome graphs pose a real challenge for alignment algorithms. The most common sequence alignment approaches use seeds as a starting point to narrow down the search space for each sequence fragment. The seeds are a fast and easily accessible representation of the available sequence space and are stored in pre-constructed indices. Such indices are often k-mer based. (Xin *et al.*, 2016), and the construction of k-mers from linear sequences is a trivial task as there is only one possible k-mer starting at each base-pair. This drastically changes in a graph context where variation gives rise to multiple k-mrs starting at the same position. Depending on the length of k and variation frequency the number of possible k-mers can make the construction of full k-mer indices infeasible, if not computationally impossible. In order to create a k-mer index from a graph this complexity needs to be reduced. This can either be done by choosing a less complex graph construction method, by removing variation from the graph in the indexing phase (*pruning*), or by considering only the linear path in the graph. Graph pruning removes nodes and edges from highly connected regions of the graph in an effort to reduce complexity. This removes information from the graph and can even cut the graph into unconnected components that are removed if they are too small. As such it can make complex regions completely inaccessible for seeding, and

therefore read mapping. This introduces a new type of reference bias. One example of such an index is the *GCSA2* index employed by *vg map*. This index is based on representing the genome graph as a *de-Bruijn* graph (Sirén, 2016). An alternative approach that does not alter the graph structure but limits the available variation by linearizing the sequence space and reducing it to the input path stored in the graph. An example of this index is implemented in *vgs GBWT* (graph positional Burrows-Wheeler transform) index. Both indices reduce the usable sequence space. While the *GCSA2* index alters the graph and can result in inaccessible highly variable regions, the *GBWT* index reduces the possible choices in the graph to those that are represented by path. Both of these index construction methods are part of the *vg* toolkit (Garrison *et al.*, 2018), which is heavily biased towards vcf-derived graphs. Therefore they struggle with highly connected, cyclic graphs. This problem is reduced in long read alignment, as the seeds do not need to cover the whole genome, but can anchor the reads to seedable regions and have them expand into the complex parts in a local realignment step (Rautiainen and Marschall, 2019; Ma *et al.*, 2022). As a result very large, and complex graphs still pose a severe challenge for graph alignment algorithms. This can be bypassed by using traditional alignment algorithms to align the sequences to the linear genomes that have been used to construct the graph and injecting the alignments into the graph paths based on their positions in the linear sequence. Similar to the GBWT index based approaches this also limits the allele combinations to those represented in the input genomes.
Even with the current shortcomings graph based alignment methods have been shown to improve the results, or simplify workflows, compared to traditional linear reference based methods. Alignments to graphs have enabled the genotyping of large structural variants on a population scale (Sirén *et al.*, 2020), and have been shown to improve the variant detection accuracy (Crysnanto and Pausch, 2019). It has also greatly improved the accuracy and downstream analysis of ancient DNA fragments (Martiniano *et al.*, 2020).

1.2.4 Usability of graphs

As in linear reference based analysis workflows, aligning reads to the graph is just the beginning. In addition to improving the established analysis workflows, graphs offer the opportunity to go even further and explore previously inaccessible territory. Using an alignment based genome graph it is possible to describe variation that is stored in the graph and even easily access nested variation. The approach that is closest to traditional variant detection is implemented as part of the *vg* toolkit (Garrison *et al.*, 2018). Here the snarls in the graph are resolved in the context of the selected paths in the graph. This results in a representation that is reference based and usable by existing analysis pipelines. Variants can also be described in a reference free way, for example by *BubbleGun* (Dabbaghie *et al.*, 2021). In this case a novel data format had to be created that limits its usability. Graph topology patterns have already been used to describe substructures in cancer genome graphs (Hadi *et al.*, 2020). The better availability and resolution

of variants presented in a graph also enables improved genotyping and dedicated tools have been developed to harness this strength (Ebler *et al.*, 2020; Grytten *et al.*, 2021). These tools use vcf based representations of the variable regions and a k-mer approach to overlap reads with the graph they build internally. Methods to directly genotype, reference independent, from complex aligned graphs are still missing. Another mainstay method that is currently being transferred to graph genomes are genome wide association studies (GWAS). This analysis is used to link variation to phenotypic traits by detecting associations between the traits and variable positions in a population of individuals. Here genome graphs have the power to massively increase the accuracy as variation can be represented more truthfully and read mapping becomes more accurate. This reduces noise in variant calls and allows the detection of clearer association signals (Gupta, 2021). Using alignments to the graph, novel variant detection is possible as well. The algorithm currently implemented as a module in the *vg* toolkit (Garrison *et al.*, 2018) is based on the *freebayes* algorithm (Garrison and Marth, 2012). The individual coverage for each position is calculated and used in conjunction with information on snarls in the graph to linearize the variation in respect to predefined paths in the graph, thus resulting in reference based variant calls in the vcf format. As mentioned in subsection 1.1.2, the impact of structural variants can only be fully understood in the light of functional annotations. While no graph based gene annotation algorithm has been implemented, yet, it is possible to project existing features into the graph using their known position on a path. One of the last big steps to make graphs usable by average scientists is the visualization of its structure. In contrast to conventional reference genomes, graphs lack linearity. This makes it much harder to visualize them and grasp the result. Of course there are methods to visualize the raw graph without any alterations. They work very well on small subgraphs, but become very hard to interpret on larger, more connected sections of the graph (Wick *et al.*, 2015; Beyer *et al.*, 2019). As a consequence methods like *panache* try to enforce a linearity in the graph by defining subgraphs as blocks without PAVs (Durant *et al.*, 2021). Other methods reduce the complexity of the structure by removing information from the graph space (Gonnella *et al.*, 2019).

1.3 Pan-genomics

The increasing availability of whole genome assemblies and accurate long read sequencing data has given rise to the field of pan-genomics. This field strives to describe a species genetic potential in a more unbiased manner. The term pan-genome describes the genetic material of a set of individuals. It can in general be split into three sub groups. The core genome, which is shared between the majority, or all of the involved individuals, the private genome, that is either completely private or rare, and the shell genome, which covers all the rest and is for example shared by members of a certain sub-population, but not by others. A sub class of pan-genomes are are pan-proteomes, that focus on the coding gene space, or pan-transcriptomes, limited to the transcribed part of the genome.

Multiple pan-genomes have been analyzed and, especially in highly complex crop plants, have led to the discovery of important, unknown, alleles (Lu *et al.*, 2015; Jin *et al.*, 2016; Gao *et al.*, 2019; Guo *et al.*, 2019; Li *et al.*, 2022), as well as insights into the wider genome structure of a population (Jordan *et al.*, 2021). Some of those projects already utilized genome graphs to store and describe the pan-genome itself (Nguyen *et al.*, 2015; Serhat Tetikol *et al.*, 2021; Long *et al.*, 2021; He *et al.*, 2023). Even in clinically relevant regions in human genetics, pan-genomes have aided a better understanding of the underlying genetic causes , for example for male-infertility (Chin *et al.*, 2023). Recently the first super-pan-genome has been analyzed. This pan-genome does not only consist of individuals of a single species, but contains multiple species of the genus poplar and helped to identify genes that drive the divergence between the different species (Shi *et al.*, 2023).

Overall the use of pan-genomes has started to take hold in the scientific community and the advantages of expanded reference structures are recognized. In this thesis I aim to build a graph based pan-genome of *A. thaliana* to better represent the variation of this species. This genome graph will be used to explore the previously hidden sequences of *A. thaliana*. To accomplish this, I will show how the standard operations, like read mapping, variant calling and variant genotyping, used on linear reference genomes can be performed on a complex genome graph. While genome graphs have been used before to represent pan-genomes, previous studies either shied away from whole-genome alignment derived graphs and resorted to vcf based graphs (Sirén *et al.*, 2020; Groza *et al.*, 2021; Serhat Tetikol *et al.*, 2021; Bayer *et al.*, 2022), avoided read alignments to whole genome alignment derived graphs (Nguyen *et al.*, 2015), or used shorter, less complex species in their analysis (Hickey *et al.*, 2020). I aim to overcome this and establish a whole genome derived graph in the plant species *A. thaliana*. In addition, I will present the novel tool *panSV*, which utilizes the unique properties of a genome graph to extract variation from it without relying on a reference genome. This enables me to describe nested variation, which has been near impossible to detect before. The six *de-novo* genome assemblies used in this project will be annotated, described and compared to the reference genome. The genome annotation will be performed by *auto-ant*, a new annotation pipeline that I designed. It combines established annotation methods in a flexible framework to annotate TEs and genes in multiple genome assemblies. Finally, I will use the constructed genome graph as a template to describe the pan-genome, and pan-proteome of the six *de-novo* assembled genomes in a larger set of 840 resequenced accessions, originating from the 1001 Genomes Project (1001 Genomes Consortium, 2016). I will also use the aligned short-read sequences to call variation from the graph in an effort to use the additional power of the genome graph.

Chapter 2

Methods

2.1 Computational tools

The availability of multiple high quality, assembled genomes of a single species and the prospect of genome graphs pose previously unknown challenges to existing tools and approaches. Therefore I had to adapt and develop novel tools to support my research. Here I present the *auto-ant* annotation pipeline, and *panSV*, a novel sequence variation detection software, that were developed by me.

2.1.1 *auto-ant*

I developed an automatic and scalable pipeline to annotate genome assemblies using different types of evidence. It combines pre-existing tools to mitigate the weakness of individual approaches. This pipeline is able to annotate TEs and genes in one single run and can be used with and without supporting RNA sequencing reads. It has been designed as a modular and easy solution that on one hand allows for a hands-off annotation by just providing the fasta sequences and a reference genome, but also allows the user to customize different input files based on their specific needs. While it can be run with and without supporting RNA sequencing evidence, it always requires a reference genome and reference annotation. Depending on the provided input data the appropriate sub modules are executed and their results combined based on a weight matrix that reflects their individual trustworthiness. In a last step orthogroups, based on the transcripts of the annotated genes, the reference genome, and potential outgroups are detected to describe the pan-proteome of the annotated genomes. *auto-ant* is written as a *Snakemake* pipeline (Köster and Rahmann, 2012). This means that it is able to detect alternative intermediate files provided by the user and skip the rules to generate them. Thus for example a different alignment tool can be used, or an external TE annotation can be provided by the user. The integration of *Anaconda.org* (anaconda, 2020), as a package manager, ensures the consistent versioning of all required tools, as they are installed by the pipeline upon its first execution.

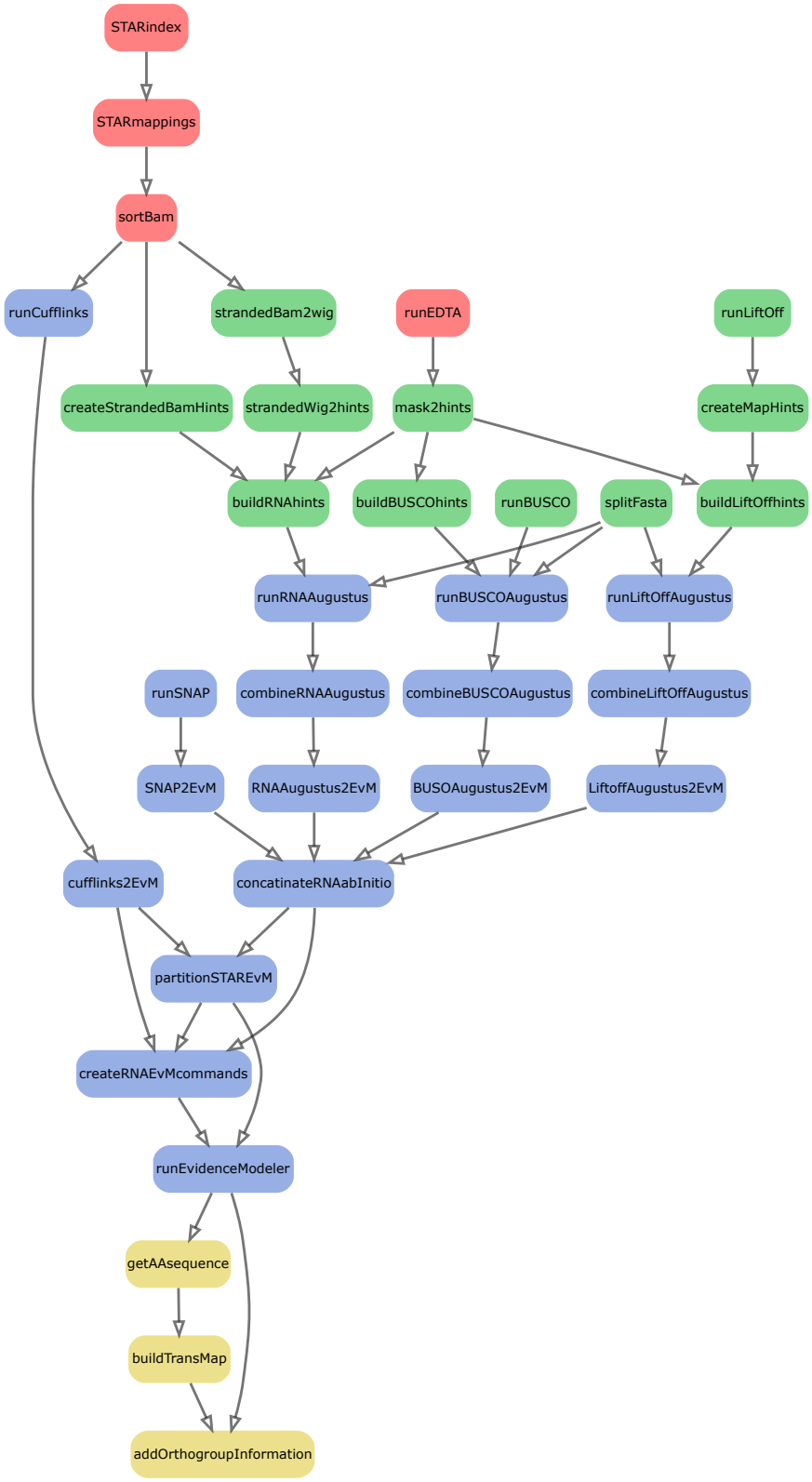

Figure 2.1: *auto-ant* rule graph - Layout of rules executed by *auto-ant*, coloured by functional groups: Data preparation (red), annotation hint and evidence generation (green), gene annotation (blue), and orthogroup assignment (yellow).

auto-ant can be divided into four major subsections (Figure 2.1). The first section is preparing the pipeline and input data. It installs all dependencies and prepares the assembled sequences, maps available RNA reads and produces a TE annotation of each genome. The second step is the annotation hint and evidence generation. In this step all annotation hints that are needed to run the annotations are prepared. This is mostly relevant for the multiple runs of the annotation prediction tool *augustus* (Stanke *et al.*, 2008). The next and largest step is the annotation itself. Here the different annotation tools are run. In order to minimize the bias of a single run and configuration independent runs of multiple tools are performed. The *de-novo* gene prediction tool *SNAP* (Korf, 2004) is run as an independent genome annotation tool, but the main annotation software used is *augustus*. Here up to three different runs are performed using different settings and input data. One is using improved training based on an updated gene model created for each input genome using *BUSCO* (Seppey *et al.*, 2019). The second run utilizes hints from an annotation lift over from a reference annotation onto the new assembly, using *ListOff* (Shumate and Salzberg, 2020). The last run is only executed if RNA sequencing data is available. This run of augustus utilizes hints created from RNA sequencing reads, mapped to the genome. In addition the mapped RNA sequencing data is assembled using *cufflinks* (Trapnell *et al.*, 2010) and provided as an independent annotation track. All independent gene predictions are weighted and combined into one coherent, final annotation using *evidenceModeler* (Haas *et al.*, 2008). The last step of the pipeline is the orthogroup assignment that calculates an ortholog based transmap using *OrthoFinder* (Emms and Kelly, 2015, 2019). Additional scripts to obtain basic statistics on the annotation and orthogroups are provided, but not part of the pipeline itself. As auto-ant produces multiple annotations in parallel, it is mandatory that the fasta entries of every assembly have unique identifiers and carry the name of their assembly in the fasta header in the format of $>$*assemblyID_contigID*. The pipeline is controlled using a config file that is used to specify all input data and settings.

Preparation step

In addition to an increase in the quality of annotations, one main objective of this pipeline is to decrease the run time. Therefore the most important preparation step is to chunk the input fasta files. As *augustus* (Stanke *et al.*, 2008) is deterministic, I can run it on individual components of the assembly and later combine them into a single annotation. The user can set a number of chunks (n $<$ number of contigs) and the pipeline will split the input fasta into *n* separate fasta files, where *n-1* files contain the *n-1* largest contigs and the *nth* file the remaining sequences. A second major step in the preparation is the TE annotation. Here *EDTA* (Ou *et al.*, 2019) is run to annotate TEs in each individual genome. This annotation will later be used to soft-mask parts of the genome in the *augustus* annotation runs. Doing so increases the accuracy of the annotation as TEs can introduce noise in the gene prediction and can result in wrong gene annotations. If RNA sequencing data is available it will be mapped to its respective genome. The fastq files

are attributed to the correct genome based on a shared file name prefix. The read mapping is performed using *STAR* (Dobin *et al.*, 2013). The maximum intron size can be set by the user. A reasonable default value for *A. thaliana* has been provided in the config file. If possible four threads are used per run and non canonical intron motifs are removed. Multiple mappings are not allowed and the two pass mode has been set to basic. The mappings are stored as bam files and merged into a single file for every accession if multiple replicates and tissues have been provided. As a last step the resulting bam files are sorted for easier access in the hint generation using *samtools sort* (Danecek *et al.*, 2021) with 12 threads.

Hint generation step

The main tool of this pipeline, *augustus*, uses hints to support, or correct, its *ab-initio* prediction. These hints are generated by a collection of rules that are attributed to this step of the pipeline. As a first step the *EDTA* TE annotation is transformed into the hints format using a custom script that marks them as regions to soft-mask in the annotation. This set of hints will be used in every run of *augustus*, independent of the other hints. *Augustus* comes with a multitude of ready to use prediction models for different species. These models are always tailored towards the species existing reference genome and thus contain a bias of their own. Therefore one run of augustus is performed with an alternative training model that is created based on the specific genome. This retraining is performed by *BUSCO* (Seppey *et al.*, 2019) for each individual genome assembly. Each retraining set is then softlinked into the species directory of *augustus*, that was installed by *anaconda*. The hints for the second run of *augustus* are created from a lift-over of a pre-existing reference annotation onto the new assembly. The lift-over is performed by *LiftOff*. Using a *minimap2* (Li, 2018) alignment of the new assembly with the reference genome, this tool utilizes the established alignment anchors to project the existing gene annotation onto the new assembly. *LiftOff* is run on default settings. The only exception is that partial hits are discarded and the reporting of multiple gene copies was activated. The resulting lift over gff file was then converted into the *augustus* hint format. If RNA sequencing data has been provided, two independent types of hints are generated using scripts provided by *augustus*. In case the RNAseq was marked as being stranded by the user the hints are generated independently for each strand and merged afterwards into one coherent hint file. The first type of hints are based on the wiggle track format that was introduced for the human genome project (Kent *et al.*, 2002). The sorted bam files are converted to wig files using the bam2wig script that is supplied by augustus. These files are then converted to hints using *wig2hints*. In most cases the default settings were used, except the minimum threshold was set to 2 and the minimum score to 4. Boundary areas with a coverage <0.1 (*–prune*) were removed and the radius of each hint was set to 4.5. The second type are expressed sequence tags (EST). They are calculated using the *bam2hints* script. The hints for each independent run of *augustus* are merged with the TE hint file.

Annotation step

The main part of the pipeline is the annotation step. Here the individual gene prediction tools are run and their results are merged into one combined annotation using *evidenceModeler*. An individual *augustus* annotation is run on each chunk created in the preparation step. The RNA evidence, and lift-over based *augustus* annotations are run using the pre-trained species model selected by the user. The additional *BUSCO* retrained model is used in a third, independent, run. *augustus* requires an extrinsic evidence matrix to weight the individual evidence tracks, based on their trustworthiness. This file can be altered by the user in order to reflect his specific needs. A file with reasonable default values for *A. thaliana* is supplied with the pipeline. Each annotation run creates an individual output file. The files for the individual chunks of an assembly are collected and merged. Next the gff3 annotations are converted into a format that can be used by *evidenceModeler* using the *augustus_GFF3_to_EVM_GFF3.pl* script that is provided by *evidenceModeler*. In order to distinguish them in the weighting process, each of the individual *augustus* annotations are marked with a unique string based on the type of evidence used. In an effort to reduce the intrinsic bias that three individual runs of the same annotation algorithm introduces, the *de-novo* annotation tool *SNAP* is run. The user has to select a species from the set of pretrained models that *SNAP* provides. Apart from this the tool is run using default settings. Again, the results of the annotation are converted by a script provided by *evidenceModeler* (*evidenceModelers SNAP_to_GFF3.pl*). If RNA sequencing reads have been provided *cufflinks* is run to add an additional layer of evidence. Based on split alignment produced by *STAR* individual transcripts are assembled, and converted into a format readable by *evidenceModeler*. *Cufflinks* is run using default settings. The conversion is performed using the *cufflinks_gtf_to_alignment_gff*2*.pl* script that is provided by *evidenceModeler*. In addition to the data type conversions *evidenceModeler* requires the input data to be chunked into 1 Mb bins with 1 kb overlap. This binning is performed by another script provided by *evidenceModeler* (*partition_EVM_input.pl*). Next the commands to combine the annotation of each bin are generated using the *write_EVM_commands.pl* script. In addition to the bins this script requires the individual annotation gff files, the assembly fasta, and a weight matrix. The weight matrix can be adjusted by the user, but *auto-ant* comes with two tested options. One that has been validated to be used with the RNA sequencing based annotations and one that works without. The weights for each input annotation were chosen based on their ability to recreate the known *araport11* reference annotation of *A. thaliana* (Cheng *et al.*, 2017). The created commands are then executed and the individual annotations are weighted based on the provided matrix to create a combined annotation in the gff3 format.

Orthogroup assignment step

The last step of the *auto-ant* pipeline detects orthologs in the *de-novo* annotations, the reference annotation, and outgroups provided by the user to link gene copies between them. The reference annotation is included to provide a framework for later comparisons and functional analysis. *Orthofinder* (Emms and Kelly, 2015, 2019) detects orthologous genes based on amino acid similarity, it therefore requires amino acid sequences. The single longest transcript of each gene is generated using *gffread* (Pertea and Pertea, 2020). The transcripts are then compared to the other annotations in a pairwise all vs all *diamond* (Buchfink *et al.*, 2021) search. A species tree of all input genomes is constructed based on the results and genes are assigned into orthogroups. The results of *Orthofinder* are post-processed to be in a tabular format that allows easier readability and also includes single genes that do not have an ortholog. In a very last step the orthogroup ID of each gene is added to the gene feature section in the annotation gff3 files for easier cross reference. If the orthogroup of a gene contains at least one gene from the reference annotation their reference gene IDs are also added to the individual gene entries.

Auxiliary scripts

auto-ant comes with a set of auxiliary scripts that are not part of the pipeline itself but can be used to gain insights into the annotations accuracy, its basic statistics and the pan-transcriptome. The *TransMapStats.py* script reads the final transmap and creates basic plottable statistics on the annotated genes, their orthogroup assignment and relationship with the reference annotation. The *TransMapRefDiff.py* script compares the exon count and the transcript length of a predicted gene with the median of its reference orthologs to estimate the accuracy of the annotation. A saturation and pan-transcriptome analysis can be performed using the *saturationAnalysis.py* script. It first calculates the core, shell, and private transcriptome based on the number of annotations that contribute genes to an orthogroup. If all annotations contribute at least one gene the orthogroup is considered as *core*. If an orthogroup only contains genes from a single annotation, it is considered *private*. All other orthogroups are attributed to the *shell* class. For each assembly the expansion or contraction of an orthogroup is calculated by comparing the number of genes this annotation contributes to the median number of genes contributed by all annotations. If the number of genes an annotation has in an orthogroup equals the median it is considered as conserved. A reference ID can be supplied using -r parameter. In this case the *core* and *shell* classification is based on the presence of the reference annotation and the copy number changes are calculated in comparison to the copy number of reference genes in this orthogroup. This script also calculates a Z-Score matrix based on the copy number changes in unconserved orthogroups. The user can either set a fixed value for missing data points, or have them be calculated as a Z-Score. Setting a dedicated value has the advantage that orthogroups with missing annotations are clearly visible in the

final matrix. The last step of this script is a bootstrapped saturation analysis. It calculates the core, shell, and private orthogroups in incremental steps for a user-defined number of random combinations. The last auxiliary script (*nonStandardOG.py*) allows the user to detect non-standard orthogroups. Those orthogroups are either unconserved orthogroups as described above, or orthgroups that have mobile genes that do not occur on the same chromosomes in every accession.

Pipeline validation

The *auto-ant* pipeline was validated using the *A. thaliana TAIR10* reference genome (Berardini *et al.*, 2015) and it's *araport11* reference annotation (Cheng *et al.*, 2017). The performance of each individual tool was measured by its ability to re-annotate the reference genome and re-create the reference annotation. The settings used in the *auto-ant* pipeline were tuned to achieve the best result for each individual tool. The performances were then compared to each other and weighted based on the reliability of the input as well as the output. This data was used to set the weights of the *evidenceModeler* algorithm. To represent RNA sequencing reads, real data originating from the *A. thaliana* accession *AT6909* were used, as the reference genome is based on this accession. The reads were mapped to the *TAIR10* reference genome using *STAR* and the same settings as in the *auto-ant* pipeline. The value for the maximum intron size was estimated using the *araport11* annotation and was set to 6,000 bp. The tools were run individually using their default settings. The resulting *de-novo* predicted features were intersected with the *araport11* annotation using *bedtools intersect* (Quinlan and Hall, 2010). Only 100% coverage of the features were considered as correct. For each feature type in the *araport11* annotation the sensitivity, as a fraction of re-annotated features fully contained in known features, as well as the specificity, as the fraction of correctly matched reference predictions, was calculated. The calculated sensitivity and specificity values of gene, CDS, and exon features were used to set the weights of the *evidenceModeler* weight matrix. The final *auto-ant* pipeline was then run to re-annotate *TAIR10* once again and the results were checked against the performances of the individual tools.

2.1.2 *panSV*

In order to overcome the established, reference focused, analysis, certain tasks need to be adapted. As variant detection is a task that is heavily influenced and biased by the reference genome I decided to develop the reference agnostic variant description tool *panSV* to use it in the analysis of the *A. thaliana* graph based pan-genome.
panSV works under the assumption that a species' pan-genome consists of a library of sequences that are either present in or absent from each individual genome. It leverages the fact that this library is already stored in the genome graph and describes such variable regions in the graph. Each variable region is described in the context of every genome that contributes to it, and thus eliminates the need for a reference genome as sole coor-

dinate space. Therefore *panSV* requires paths to be present in the graph. It is built to detect variation of any size, from SNPs to chromosome arm size alterations including their nested variation and describe them in a hierarchical format. The hierarchical format enforces a strict parent-child relationship of nested variation that is reflected in the naming scheme of the reported variable regions. With the idea of a library of variable sequences in mind *panSV* is based on the concept of core levels. The core level of a sequence in the graph is defined as the number of genomes that contain this sequence. With the help of this concept *panSV* can identify, and describe, regions of the graph where the core level varies. This concept results in regions with properties that can differ from that of a classic bubble structure in that it does not need to be a closed subgraph. *panSV* describes the de- and increasing core levels of the graph, independent of the paths that are the cause for that change in core level. This means that multiple regions can share a subset of nodes, without being direct parent, and children. These variable regions then form a sibling relationship. The output of *panSV* is an easily comprehensible BED file alongside additional statistics files that represent the nested structure and explain basic statistics of bubbles and traversals.
Currently two versions of this algorithm exist. The early version, implemented by me in python, will be used in this thesis. A second version, that massively increased performance has been implemented by Sebastian Vorbrugg in Rust, will remain under active development.

panSV algorithm

panSV uses the *gfautils* library (Kubica, 2021), which I created to read, store, and access a graph in GFA format. After reading and processing the graph, a core level is calculated for each node in the graph. The core level is the sum of genomes that traverse a node, where each genome can have multiple paths, but only contributes once to the core level of a node. Differences in the core level are used to detect variable regions in the graph. They are defined by two anchoring nodes with a higher core level than the nodes that are located between the anchors. The paths that go through such a variable region form traversals that can be used by multiple paths.
In order to detect the variable regions, *panSV* traverses each path from start to end and compares the core level of two consecutive nodes. If the core level decreases a new variant traversal is being started and the previous node is saved as the left anchor, together with the current position in the path. The current variant traversal is kept open until a consecutive comparison shows an increase in core level to at least the level of the left anchor. The current path position is considered the stop position of the traversal and the node becomes the right anchor of the variable region. The two anchors are then compared to a list of already known anchors. If a variable region with these anchors already exists the current traversal will be added to it. The list of nodes from the current traversal are compared to the nodes of previously observed traversals of this region. If the same traversal has already been observed the current path is added to it. Otherwise

a new traversal is added to the region. If no region with the anchors exists a new one is created and the current traversal stored as its first traversal. Next all recently closed regions with a lower core level are compared to the node set of the current region. If they are a 100% subset, they are saved as sub-regions, representing variation nested within the closed traversal. In the next phase *panSV* detects PAV type bubble traversals by re-walking each path and searching for instances where the two anchors of a known region are directly adjacent in the path. They are treated as traversals of this region.

Algorithm 1 *panSV pseudo-code example*

```
1:  for path in graph do
2:    for node in path do
3:      if node.coreLevel > nextNode.coreLevel then
4:        add node to all open traversals
5:        start new traversal
6:      else if node.coreLevel < nextNode.coreLevel then
7:        add node to all open traversals
8:        close all traversals with coreLevel ≤ nextNode.coreLevel
9:        if region with anchor nodes of the closed traversal exists then
10:         add closed traversal to existing region
11:       else
12:         create new region with anchors of the closed traversal; add current
            traversal
13:       end if
14:     else
15:       add node to all open traversals
16:     end if
17:   end for
18: end for
end
```

The definition of variable regions implemented in *panSV* does not follow the strict defintion of bubbles. Therefore multiple variable regions can share a subset of nodes. In order to accommodate this *panSV* searches for sibling regions by overlapping the node sets of variable regions. If two variable regions of the highest core level share a subset of nodes they are marked as siblings in the output.
In the last step an ID is assigned to each variable region. The IDs are made from '.'-separated numbers. The number of sparations depend on the number of core levels in the graph and are used to show the parent-child relationship of parent and child regions. Starting from the highest core level each variable region is assigned a consecutive numerical ID, starting from 1. A child inherits the ID of its parent and its own numerical ID is added to the corresponding core level of the ID. For example a parent ID of a core level 5 graph would be 5.0.0.0. The first direct child would be assigned the ID 5.1.0.0,

if it is a core level 4 region, or 5.0.1.0 if it is a core level 3 region. Using this naming convention the nested relationships are encoded in the IDs and can be interpreted easily. In addition each traversal of a variable region is also assigned a numerical ID.

panSV output

By default *panSV* creates three different output files. A BED file, detailing the positions of each traversal in the individual genomes coordinate system, a traversal file that contains information of the traversals and their paths, and a statistics file with basic statistics on each variable region. In addition an optional sibling file can be obtained that contains the sibling region IDs for each detected variable region.
The BED file contains the position of the variable regions in the coordinate system of the individual paths of the graph. It is a combined file for all paths in the graph, but can easily be split based on the path IDs. The first three columns are that of a standard BED file. The fourth column holds the ID of the variable region that is separated by a '_' from the ID of the traversal this path uses to traverse this region. The fifth, and last, column contains the core level of this region.
Basic statistics for each variable region are stored in the statistics file. They are connected to the region IDs. For each region this file contains the core level, the number of subregions, the combined length of all nodes that are part of this variable region, as well as the minimum, maximum and average traversal length. It states the number of traversals through the region, as well as the number of paths that traverse them. This number can be higher than the core level if paths from the same sample traverse a variable region multiple times. Lastly, if the corresponding option was set, it contains the number of siblings that this region has.
The traversal file stores information on each traversal. It contains the traversal ID, followed by the region ID this traversal belongs to. It also contains the length of the traversal, the number of paths that traverse it and the sum of all traversals through it. This number can differ from the number of paths if a path traverses it multiple times. In addition it also contains a ';' separated list of all path IDs that use the traversal.
The optional sibling file has two columns. The first contains the region ID, while the second contains a ';' separated list of all siblings detected for it.

2.2 The sixRef project

In order to step away from a singular reference space and move towards a pan-genomic view of species and genomes we need to enlarge the sequence space. This requires additional genome assemblies to construct the graph and analyze the pan-genome represented by it. For this purpose six accessions of *A. thaliana* were chosen to be sequenced and *de-novo* assembled (Table 2.1). The accession which the current *TAIR10* reference genome (Berardini *et al.*, 2015) is based on, *AT6909*, was chosen, together with the widely used mutant *AT7213*. Four additional accessions were chosen to complete the set based on their sequence, and geological divergence (Figure B.1), which was estimated during previous resequencing experiments with the *TAIR10* reference genome. During the assembly phase it was discovered that the accession *AT7186* had initially been mislabeled as *AT7063*. The assembled genomes were annotated for genes, TEs, and repeats. The gene annotations were combined in an orthogroup assignment in order to describe the pan-proteome. In addition the genomes were aligned to the *TAIR10* reference genome and variants were called in a reference based framework. In a last step a genome graph was built that was used to describe the pan-genome stored in it and served as an alignment target for a large collection of short-reads from the 1001 Genome Project (1001 Genomes Consortium, 2016) to show the feasibility of such an approach, genotype the pan-genome in the larger population, and call novel variants.

Table 2.1: Admixture groups - Accession ID, name and admixture group of the six accessions selected for *de-novo* assembly as recorded in the 1001 genomes project.

Accession ID	Accession Name	Admixture Group
AT1741	KBS-Mac-74	germany
AT5784	Ty-1	admixed
AT6909	Col-0	central Europe
AT6911	Cvi-0	relict
AT7186	Kn-0	central Europe
AT7213	Ler-0	admixed

2.2.1 Assembly

The sequencing data generation and genome assembly was performed by Dr. Felix Bemm. The seeds for the six accessions were taken from the internal seed stocks of the Ecker and the Weigel labs. The plants were grown at 20°C in a growth room with a 13h daylight phase. Leaves were harvested approximately three weeks after bolting. DNA was extracted and prepared for sequencing. For each accession three types of genomic sequencing data was produced. *PacBio* long reads, 250 bp PCRfree paired-end *Illumina* reads, and an optical map for scaffolding. The assembly of the PacBio long reads was performed using *canu* (Koren *et al.*, 2017). The expected genome size was set to 140 Mb. Two separate error correction and polishing steps were performed on the assembly. The first used the original long reads and *quiver* (Chin *et al.*, 2013), while

the second leveraged the lower error rate of the short-reads and was performed by *pilon* (Walker *et al.*, 2014). The contigs were scaffolded using optical maps and possibly misassembled regions were identified by an previously established python based SV-calling pipeline (O'Neil, 2016). Erroneous regions were corrected using the results. The length cut-off for this step was set to 5kb, as the low resolution of optical maps can result in false positive calls (Kawakatsu *et al.*, 2016). During the scaffolding step the gaps were filled with Ns, according to their size, estimated by the optical map alignment. Statistics on the assemblies were collected by me using the *SeqFilter* toolbox. The scaffolded chromosomes of the individual assemblies were aligned with the *TAIR10* reference genome (Berardini *et al.*, 2015) using *minimap2* (Li, 2018) with the *asm5* preset. In order to examine high level structural variants the resulting alignments were visualized as dot plots with the help of *minidot* from the *miniasm* package (Li, 2016). Finally I ran *BUSCO* (Seppey *et al.*, 2019) to assess the completeness of the genome assembly. The *embryophyta_odb10* data set was downloaded using the internal download method of *BUSCO* and was used in the analysis.

2.2.2 SV calling & evaluation

In addition to conventional short-read variant calls, assembly based methods have become a viable option to detect longer variants that have been hard to detect using short-reads. While conventional pairwise alignments were the method of choice so far, graphs are emerging as an option to represent multiple whole genome alignments and to be used for variant detection. I called variants based on the new assemblies using a pairwise approach and a graph based approach. In addition I created a subset of the six accessions from the full 1001 Genomes variant calls (1001 Genomes Consortium, 2016). The three results of the three approaches were first analyzed individually and then intersected and compared. After generating each set of results they were submitted to the same post-processing pipeline.

Short-read based variant calls

For the comparison and validation of different variant calling approaches I extracted the variants of the six assembled genomes from the set of variants called in the 1001 Genomes Project. These variants were called by mapping short-read sequencing data to the *TAIR10* reference genome (Berardini *et al.*, 2015). I downloaded the dataset (*1001genomes_snp-short-indel_only_ACGTN.vcf.gz*) from the 1001 Genomes web page (*https://1001genomes.org/*). I subsetted and recoded the file to only contain the variants of each of the six accessions using *vcftools* (Danecek *et al.*, 2011). Each individual vcf was then submitted to the post-processing steps described below.

Pairwise alignment based variant calls

Pairwise whole genome alignments can improve variant calls as they enable a better resolution of large structural variation that is otherwise hard to detect using short-read based methods. *SyRI* (Goel *et al.*, 2019) is a tool that can call variants from such an alignment. This software requires a one-to-one relationship between aligned scaffolds, therefore I had to remove the unplaced contigs of each assembly. Only the chromosome scaffolds were used in the alignment. I ran pairwise genome alignments for each genome assembly against *TAIR10* using *minimap2*. Due to the low sequence divergence between the *TAIR10* reference genome and the new genome assemblies the *asm5* alignment preset was used. In addition the *-eqx* parameter was used to include =/X CIGAR operators in the sam output file. This file was then used by *SyRI* to detect the synteny structure as well as small and large variation between the two genomes. *SyRI* was run on default settings and all intermediate files were kept for later use. I split the output vcf file into two separate files. A bed format file containing the large structural variants as marked by *SyRI* with '<..>' variant identifiers. These variants were further classified as syntenic sequence, rearranged reference sequence, or novel sequence, based on the class assigned by *SyRI*. This file was merged with the set of unaligned sequences, using reference positions. The remaining variants were kept in the vcf format. By using alignment coordinates provided by *SyRI* I also created bed files that hold the locations of structural variants in the non-reference coordinate space. As structural variants can contain nested variation I intersected them with the small variant vcf file using *bedtools intersect*, and calculated the variant number per kilo base for each class of structural variant. The structural variants reported by *SyRI* were visualized using *plotsr* (Goel and Schneeberger, 2022).

Graph based variant calls

In contrast to the reference biased variant calls of the 1001 Genomes Project and the pairwise variant calls by *SyRI*, the genome graph that I constructed (subsection 2.2.4) contained the complete sequence of all six assemblies and the reference. This enables a more truthful representation of the genetic relationship of the accessions. I called reference based variants from the graph using *vg deconstruct*. All paths of the *TAIR10* reference genome were used as coordinate system. The output vcf file contained every path in the graph as individual sample line. I used the accession IDs to combine samples into a single sample column per accession. In addition to the joined file I created an individual file per accession. All vcf files were then submitted to post-processing.

Variant post-processing

The vcf files of each variant detection approach were submitted to the same post-processing steps. If necessary vcf files were merged using *vcftools vcf-merge* (Danecek *et al.*, 2011). Heterozygous calls were split using *vcflib's vcfbreakmulti* (Garrison *et al.*, 2021), sorted

using *vcftools vcf-sort*, compressed using *bgzip*, and then indexed using *tabix*. I categorized the variation in each file into one of three different classes based on their size. *SNPs*: if one base pair was replaced by another single base pair. *Small variants*: for everything smaller than 50 base pairs and *large variants* for all variations bigger than 50 base pairs.

Variant intersection

In order to describe the differences in variation called by each approach I intersected the variants detected by the different methods. Each variant class was intersected individually using *bcftools isec –nfiles +1* (Danecek *et al.*, 2021) in order to get a presence-absence matrix. This method only returned perfect matches. All *small* and *large variants* that were not intersected were converted into bed files and a non-perfect intersection step with the vcf files was performed in order to find overlaps with smaller variants using *bedtools intersect* (Quinlan and Hall, 2010). I reported the number of *small* and *large variants* that overlapped with at least one variant in the vcf file, as well as the number of variants contained in each variant.

2.2.3 Annotation

An annotation of *de-novo* assembled genomes is a vital step towards a better understanding of the pan-genome and pan-proteome of *A. thaliana*. Therefore I ran the new *auto-ant* annotation pipeline to predict TEs and genes in each assembly and created an orthogroup based trans map. In addition I identified repeats in the sequence using *RepeatMasker* (Smit, AFA, Hubley, R & Green, P., 2013).

Gene annotation

I used the *auto-ant* pipeline described in subsection 2.1.1 to annotate genes and TEs in the assemblies. In preparation for the annotation RNA sequencing evidence was produced by Felix Bemm. He performed an RNA sequencing experiment for four different tissues, or developmental stages (flower, leaf, root, seedling) for each accession. The RNA was extracted from fresh plant material using the *RNeasy Plant Mini Kit* in three biological replicates per tissue. The libraries were prepared using the protocol described by Kawakatsu (Kawakatsu *et al.*, 2016). The completed libraries were sequenced as 150 bp single end reads on an *Illumina HiSeq2500* instrument. The sequenced reads were pre-processed by me. As the adapters used for the sequencing were unknown and could not be recovered or removed by standard trimming tools a first pass of quality trimming was performed using *cutadapt* (Martin, 2011). I trimmed Ns off the ends of each read and reads shorter than 100 bp were discarded. As a next step the first 10 bases were cut off each read using *awk* to get rid of the adapter. The processed reads were then fed into the *auto-ant* pipeline to perform the annotation. I ran *auto-ant* in RNASeq and

stranded mode. Based on the presence of six chromosomes and a set of unplaced contigs in each assembly I set the chunk number for the contig splitting to 7. In the read mapping step with *STAR* (Dobin *et al.*, 2013) the maximal intron length was set to 6 kb. TEs were detected using *EDTA* (Ou *et al.*, 2019), independent of the pipeline. *EDTA* was run on default parameters, except it was provided with the *araport11* (Cheng *et al.*, 2017) coding sequence fasta file to mask those regions for the TE annotation. In addition the options to run *RepeatModeler* (Smit and Hubley, 2008) and to perform the whole genome TE annotation were enabled. The annotation gff files were provided to *auto-ant* for hint generation. The *TAIR10* reference genome (Berardini *et al.*, 2015) and the *araport11* annotation were provided for the liftover of known gene models onto the *de-novo* assemblies and the later orthogroup assignment. *BUSCO* was run using the *embryophyta_odb10* database. The training sets for *A. thaliana* were used in every appropriate tool of the pipeline. In the orthogroup assignment step two additional coding sequence files of were provided. For both annotations only the single longest transcript of each gene was used. One annotation was the *araport11* reference annotation to enable me to place the *de-novo* annotated genes in the known reference framework and easily assign potential functions. The other annotation was an *Arabidopsis arenosa* annotation, used as an outgroup (Cristina Barragan *et al.*, 2021). Using an outgroup improves the performance of *Orthofinder*. The pan-transcriptome statistics of the annotation were calculated by running the *saturationAnalysis.py* script from *auto-ants* auxiliary script collection. Orthogroups in the final Transmap were filtered to identify non-standard orthogroups using the *nonStandardOG.py* script.

Repeat detection

Repetitive regions in the genomes were annotated using *RepeatMasker* (Smit, AFA, Hubley, R & Green, P., 2013). The software was run in sensitive mode with the pre-trained *A. thaliana* species model. The results were combined with the 'repeat_region' entries in the *EDTA* gff file to create a joined repeat database. Those lines were then removed from the *EDTA* TE gff file. I then intersected the two files using *bedtools intersect* (Quinlan and Hall, 2010) and removed all repeat entries that overlapped with TEs to create two non-overlapping sets.

2.2.4 Graph construction & processing

The genome graph was constructed using the *pggb* pipeline (Garrison *et al.*, 2023). This pipeline runs a multi step process to convert fasta input sequences into an aligned genome graph. It first aligns the input sequences, then converts the alignments into a graph format and finally topologically sorts, smoothes, and realigns the graph. In my graph construction I aim for a connected graph that allows multiple copies of repeats and TEs to be aligned. Meanwhile the graph should maintain the syntenic linearity of the input sequences. I evaluate the alignment represented by the graph, by extracting and analyz-

ing the regions of the graph that were not aligned to the known reference genome. In addition I use the *panSV* software, developed here, to extract variable regions from the graph and describe them using the annotations of the six *de-novo* assembled genomes.
The graph is induced based on an all-vs-all whole genome alignment performed by *wfmash* (Marco-Sola *et al.*, 2021). *wfmash* was run with a segment length of 10 kb and a block length of 30 kb. The mapping identity was set to 90% and based on the results of the orthogroup assignment and TE annotation up to 40 secondary alignments were allowed to align high copy number genes and TEs. All other settings remained in their default state. The resulting paf-alignment file was then fed into *seqwish* (Garrison and Guarracino, 2023) to be converted into a genome graph in gfa format. Here almost all of the default settings were used, with the following exceptions. The minimum match length was set to 19 bp and the transitive batch size was set to 10 Mb. As the first graph construction can contain partially unaligned regions as well as overly connected subgraphs consisting of very small nodes the *seqwish* graph was submitted to a finishing step using *smoothxg* (pangenome consortium, 2023). The minimum edit based identity of blocks was increased to 0.7 and the alignment score parameters of the partial order alignment were changed to be more conservative in an effort to prevent overalignment of repetitive sequence motifs. The updated alignment score parameters were 1,19,39,3,81,1 (default: 1,4,6,2,26,1 [match,mismatch,gap1,ext1,gap2,ext2]). The final graph was then converted into the *vg* format using *vg convert -v* and nodes longer than 1 kb were split by *vg mod -X 1000*. Afterwards the resulting graph was converted back into gfa format using *vg view*. The resulting gfa file as used in all downstream steps of this thesis.

Graph based pan-genome

The graph allows me to evaluate the pan-genome as described by the aligned accessions. As part of this I performed a saturation analysis that classified the nodes in the graph as *core*, if all accessions traversed the node, and *shell*, if only some traversed it. This analysis was bootstrapped and in each step every possible combination was calculated. In addition the sequence alignment rate was calculated for each accession in the graph. This was done from a true pan-genomic standpoint, considering the *core* genome, as well as from a reference based point of view. On the basis of the pan-genome alignment, sequences could be classified into three categories: *core* sequence, across all seven assemblies in the graph; *private* sequence, sequence that cannot be aligned to any other genome; and *shell* sequence for the remaining sequence. The sequence length as well as its fraction in each genome were calculated for each category. For the reference based alignment analysis the sequences were also divided into three categories: *reference* sequence, if the reference was aligned to this sequence, *aligned* sequence, if the sequence was aligned to at least one other accession, but not reference, and *private* sequence, if the sequence was unique to one accession. Again the sequence length and its fraction as part of each assembly, and the full graph were calculated. I also collected statistics on every node in the graph. For each node I recorded its length, the number of paths traversing it,

its status in the reference and non-reference pan-genome and a binary identifier if it is a 'repeated' node. Here, 'repeated' means that at least one accession traverses this node multiple times. As a last value all paths that traverse this node at least once are stored in a ";" separated list. This data was then used to identify nodes of interest and further investigate them.

Non-reference sequence

In order to assess the quality and compactness of the graph I analyzed the parts of the graph that were not aligned to the *TAIR10* reference genome (Berardini *et al.*, 2015). These non-reference sequences were compared to the reference based variant calls by *SyRI* (Goel *et al.*, 2019). In addition I performed a taxonomy analysis to detect their potential ancestry using *Kraken2* (Wood *et al.*, 2019). I altered the *panSV* algorithm to extract non-reference sequences. It now traverses each path, but unlike the original algorithm it does not break at changes in the core level, but at locations where the path deviates from a reference path. The detected traversals are reported in a bed format that includes the traversed nodes and full sequence. For the analysis I removed all detected non reference sequences smaller than 50 bp and all sequences that contained more than 90% Ns. I first intersected the variants with the not-aligned sequences as called by *SyRI*. The remaining variants were intersected with the remaining *SyRI* variants. The intersections were performed using *bedtools intersect* (Quinlan and Hall, 2010). A 90% overlap of the variants with the non-reference sequences from the graph was required. The same analysis was repeated for the annotated features. The positions of the non-reference sequences in each assembly were intersected with the previously annotated features: genes, TEs and repetitive regions. Each had to be contained to at least 90% in the non-reference region. The ancestry analysis was performed using *Kraken2*. I used the pre-compiled *Kraken2* plant database and augmented it with additional, more recent full genome assemblies of other plants to obtain a better representation of the close relatives of *A. thaliana*. I added assemblies of *A. arenosa* (Cristina Barragan *et al.*, 2021), *S. lycopersicum* (Budiman *et al.*, 2000; Wang *et al.*, 2005), *B. vulgaris* (Mitchell , Mitch), as well as unpublished genome assemblies of *A. lyrata*, *C. bursa-pastoris*, *C. rubella*, and *C. orientalis* (Table 2.2). The taxonomy structure was downloaded using *kraken2-build –download-taxonomy* and the sequence database using *kraken2-build –download-library plant*. I used the process described in the *Kraken2* documentation to add the additional assemblies. The final taxonomy database was then constructed with *kraken2-build –*

Table 2.2: *Kraken2* species - Additional assemblies added to the custom *Kraken2* database.

Species	BioProject
A. arenosa	*PRJEB42625*
Arabidopsis lyrata	in house
Capsella bursa-pastoris	in house
Capsella rubella	in house
Capsella orientalis	in house
Solanum lycopersicuma	*PRJNA119*
Beta vulgaris	*PRJNA413079*

build. *Kraken2* was run for each set of large non-reference sequences ($\geq$ 50 bp) with the new database. In addition to the standard output I requested the report file using *–report*. The taxonomy was visualized with *pavian* (Breitwieser and Salzberg, 2020).

Pan-genome based variant detection

While conventional variation detection approaches link variation to coordinates in a linear reference, the graph has the unique property to also explain the nested variation and population structure of the variants. In order to access this variation I developed the *panSV* algorithm. This tool enables us to describe the variation stored in the graph in an unbiased, reference free way. I used this software to extract the variants stored in the genome graph of the six *de-novo* assemblies and the *TAIR10* reference. I ran *panSV* on the full gfa graph to obtain the variable regions. An individual bed file for each of the six assemblies was created. These files were then intersected with the set of non-standard orthogroups identified by *auto-ant* using *bedtools intersect* and required at least 90% overlap of a gene with the variable region. I also classified the variable region detected by *panSV* into *SNPs*, *small variants*, and *large variants* based on their minimum and maximum traversal length. A variable region was identified as *SNP*, if all traversals had a length of 1. It was called a *small variant*, if the largest traversal was no longer than 50 bp. Everything else was labeled as *large variant*.

2.2.5 Graph alignment evaluation

In order to describe and genotype the distribution of non-reference sequences in the graph, I aligned short-read data to the graph. This required an evaluation of the performance of the available graph alignment algorithms on different target graphs. I constructed a set of five different genome graphs, of increasing complexity, and used four different alignment tools to align short-reads to them. I selected two alignment tools from the *vg* toolkit (Garrison *et al.*, 2018) (*vg map* and *vg giraffe*). While the *vg* team had already tested the performance of their mapping algorithms on simulated data and vcf based graphs, and applied them successfully on real life data (Sirén *et al.*, 2020) I want to map reads to complex graphs. It is known that the *vg* alignment tools can struggle with high complexity graphs, therefore I decided to add two additional alignment approaches to the analysis. The first is *graphAligner* (Rautiainen and Marschall, 2019), an alignment tool initially designed for long read alignments. At the time of analysis no other alignment software was publicly available, therefore I decided to try it on short-read data. The last approach I decided to test is a very recent addition to the *vg* toolkit. This approach injects mapped reads from a bam file into a graph structure based on their alignment positions in a linear sequence. Here I mapped reads to a concatenated version of all input genomes using *bwa mem* (Li, 2013) and injected them into the alignment based graphs. This has the advantage that an established alignment algorithm can be used, but limits the accessible allele combinations to those in each of the flat sequences.

I selected 12 short-read sets to be mapped to the five different graphs. Six of them were the 250 bp PCRfree reads used to polish the *de-novo* assemblies. The other six were randomly selected from the 1001 Genomes Project (1001 Genomes Consortium, 2016) short-read data and were required to have a N90 read length of over 100 bp and an estimated genome coverage of at least 10x. The five different graphs were constructed with increasing complexity using different approaches. Each graph was first converted into a *vg* format using *vg convert*, nodes longer than 1000 bp were split into multiple nodes using *vg mod -X 1000*. Afterwards the *vg* graphs were converted back to gfa format using *vg view*. The different graphs are:

Flat graph: The flat graph was constructed from the *TAIR10* reference genome (Berardini *et al.*, 2015) using *vg construct* without a vcf file. This graph was used as a baseline to compare the individual mapping performances with *bwa mem*.
VCF graph: This graph was built by inserting variation from a pre-computed vcf file into the *TAIR10* reference genome (Berardini *et al.*, 2015). I used *vg construct* with the *TAIR10* reference genome and a vcf file, obtained by running *vg deconstruct* on the complex graph. I ran the variant calling with all *TAIR10* paths as reference paths and grouped all paths from an individual accession into a single sample column. As a result this graph lacks all relationships between multiple copies of sequences and nested variation.
Chromosome graph: This graph was built by separating all chromosomes and aligning them individually to their counterparts in the other accessions. The unplaced contigs from the six new reference assemblies were aligned with the chloroplast and mitochondrial genome of the *TAIR10* assembly (Berardini *et al.*, 2015). The graph was constructed using the *pggb* pipeline (Garrison *et al.*, 2023) using settings that enforced strong linearity in the graph. The segment length was set to 20,000 bp and the block length to 60,000 bp. The mapping identity was increased to 95% and the number of secondary matches was set to 7. The individual chromosome-wise graphs were converted into *vg* graphs using the settings described above and then joined together using *vg combine*. This graph is capable of representing repeat structures and translocated sequences in their genetic context on each chromosome. In addition, interchromosomal connections are impossible.
Linear graph: The linear graph was constructed using the same settings as the chromosome graph, but instead of splitting the chromosomes, all sequences were aligned in one run of the *pggb* pipeline. While still enforcing a strong linearity this graph is capable of having interchromosomal connections.
Complex graph: The last graph was the sixRef full genome graph constructed for this project. *pggb* was run with settings that are described in subsection 2.2.4. This graph allows the collapse of a large amount of related sequences and compresses the input genomes in the graph structure.

In order to align reads to the individual graphs, indices had to be created. I ran *bwa index* on the *TAIR10* reference genome for the baseline mappings, as well as on a fasta file that contained all genomes used in the graph construction processi This index will be used for the alignment projections by *vg inject*. *vg autoindex* was used to construct the indices required by *vg giraffe* for each individual graph. The xg index for the downstream analysis of the *vg giraffe* mappings was built from the giraffe.gbz graph. The indices required for the *vg map* alignments were also constructed using *vg autoindex* for the *flat graph* and the *VCF graph*. The *pggb* based graphs required pruning steps in order to construct the gcsa index. I ran *vg prune* to reduce the graph complexity before index construction. I first removed all nodes with a degree over 3 and all resulting subgraphs shorter than 48 bp. I also reduced the k-mer length to 18, and removed all k-mers with more than 3 edges. The gcsa indices were then constructed by *vg index* on the pruned graph. The xg indices were constructed from the unpruned graphs and were also used in the evaluation of the *graphAligner* and *vg inject* based alignments. Every alignment tool was run with 12 threads and on default settings unless specified. The *bwa mem* mappings to be used by *vg inject* were run with the *-a* option to output all valid alignments, and *-q* to prevent the mapQ modification of supplementary alignments. The mappings were converted to bam format on the fly and then projected onto the individual graphs using *vg inject*. *vg map* and *vg giraffe* were run on default settings without any adjustments. *graphAligner* was run using the *vg* preset (*-x vg*). The computational resources that each alignment used were recorded with. */usr/bin/time -v*. The system time, as well as the maximum resident memory were reported for each run. The system time of *bwa mem* and *vg inject* were combined and the maximum memory consumption kept. The number of sequences aligned to each graph were collected using *vg stats*, or *samtools stats*. On graphs the number of 'total aligned' reads was stored and on the flat references the number of 'reads mapped'. I recorded the number of covered bases for each alignment using *vg pack -d*, or *samtools depth*.

2.2.6 Graph genotyping

While we previously had to rely on the sequence present in the *TAIR10* reference genome (Berardini *et al.*, 2015) to anchor variation, I can now use the sequence stored in the genome graph to improve read mappings and quantify the frequency of previously inaccessible variation using the available short-read sequencing data. I employed the combination of *bwa mem* and *vg inject*, as described in subsection 2.2.5 to map short-read data sets from the 1001 Genomes Project (1001 Genomes Consortium, 2016) onto my constructed genome graph. I described the alignment statistics, estimated the individual genome sizes using coverage in the graph, and analyzed the remaining unmapped reads. I then used the mapping coverage on the graph to describe the frequency of annotated TEs and orthogroups to gain further insights into the pan-genome. In the last step I called reference based variants and compared them to the original 1001 genome calls to describe the differences and possible improvements of the graph method.

Read mapping

For the graph alignment I used a subset of the *A. thaliana* 1001 Genomes short-reads (1001 Genomes Consortium, 2016). I removed read sets with a N90 read length below 100 bp and an estimated reference coverage below 10x to ensure I had sufficient reads as well as read length to anchor them reliably to the genome graph. The 1001 Genomes Project short-reads had been pre-processed before submission to the archives and thus I was able to omit this step. The reads were mapped to a concatenated version of all genomes in fasta format using *bwa mem -a -q*. This allowed for multiple mappings of individual reads in the shared regions of the genomes. The unmapped reads were extracted with *samtools view -f 4* and converted to fasta format for further processing. The mapped reads were then projected into the graph structure by *vg inject*. This step translates positional information of flat reference mappings into the graph space based on the shared path in the graph. The resulting gam file was used to extract the per base and edge coverage using *vg pack*. The covered edges were extracted with *vg pack -D*. Uncovered edges were removed from the output to minimize the storage footprint. The per base coverage (*vg pack -d*) was processed using a custom python script (*sum_coverage.py*) that returned the covered regions in a bed format and a matrix of covered and uncovered nodes. For both outputs I calculated the median coverage and excluded the lower 5% of the coverage distribution to remove spurious alignments. Nodes had to have at least 80% of their bases covered, with a coverage above the 5% cutoff, in order to be considered as covered in the matrix. I defined blocks of covered sequence by combining all adjacent bases with a coverage above the lower boundary and calculated their median coverage. Two independent, adjacent coverage blocks were combined into one if they were closer together than 10 bp. In such a case the calculated median coverage included the coverage of the gap bases. I estimated the repetitiveness of a coverage block by describing it as a multiple of the median coverage. The coverage blocks were reported in a bed style format using the node IDs as sequence IDs.

Genome size estimation

The coverage of each read set in the graph was estimated based on the node coverage matrix. I summed up the size of all covered nodes. The graph coverage was correlated with the admixture group as well as the laboratory that sequenced this read set. In order to estimate the full genome size I created a 19-mer index of the unmapped reads with *jellyfish*. I excluded k-mers with a count below the 5% coverage cut off, defined in the coverage analysis, and summed the count of all other k-mers. The sum of the k-mers was divided by the median coverage to estimate the amount of sequence not represented in the graph. In addition I checked the taxonomy of unmapped reads by running *Kraken2* with its default database (Wood *et al.*, 2019).

Graph pan-genome

The individual node coverage matrices were combined into a single matrix by concatenating them. This matrix was used in the mapping-based pan-genome analysis. First I calculated the *core*, *shell* and *private* genome. Due to the large number of accessions the requirements for *core* and *private* sequences were relaxed. Nodes that were covered by at least 90% of the lines were considered as *core*, while nodes that were covered by fewer than 10% of the lines were considered as *private* (*mappingSaturation.py*). In order to estimate the power of the graph to predict the pan-genome distribution I compared the number of assembled graph accessions that traverse each node with the number of accessions that cover that node in the graph. This value was normalized by the amount of total nodes in the graph. The outliers were extracted and the nodes were intersected with the annotated features to categorize them.
In addition to the overall mapping based pan-genome I zoomed in to describe annotated features of the genomes. To do so I projected the annotations into the graph space and intersected the positions with covered nodes from each accession. In the first step I created a bed file that translated nodes into the flat sequence space of each assembly (*gfa2bed.py*). This enabled me to calculate exact positions for all features in the annotations. For TEs I intersected every feature in every accession with the node space and merged all intersecting intervals independent of their TE type and origin (*projectBed2graph.sh*). I then intersected this joined TE library with the coverage bed files obtained from the read mappings (*intersectTE.sh*). Each intersection was required to cover at least 80% of the projected feature. I calculated the change in coverage for each mapping set, compared to the median coverage of the TE nodes, as well as the total amount of TE sequence covered by each mapped accession. Genes that were projected into the graph space were not merged over all accessions, but were kept as independent bed entries with assigned gene ID and orthogroup ID to enable me to detect individual genes in the downstream analysis (*projectGFF2graph.sh*). The combined gene library was intersected with the coverage intervals (*intersectTE.sh*). A gene required 80% of its sequence to be covered. I calculated the coverage for each gene and used this information to check how many representatives of an orthogroup were covered by each short-read set. In addition I calculated the mean coverage change for each orthogroup. I used the threshold defined above to categorize the genes into *core*, *shell*, and *private* genes for each short-read accession. Using the calculated coverage change compared to the accessions median coverage I calculated a z-Score for each orthogroup and accession to discover changes in the copy number of genes in specific accessions.

Variant calls

I called sequence variants from the graph based on the existing *TAIR10* reference genome framework (Berardini *et al.*, 2015). The gam mapping files were transformed using *vg pack*. Afterwards variants were called using v*g call*. I removed all calls without

the PASS quality flag. The number of homozygous and heterozygous calls was calculated for each accession. Next I subsetted the original 1001 Genomes Project short-read calls and extracted the calls of the short-read sets used in this analysis. The number of exactly matching calls for each accession was calculated by intersecting them using the *intersectVCFs.sh* script. The remaining calls were subjected to a positional overlap to accommodate fuzzy alignment borders and different variant resolutions of the two sets (*getOverlaps.sh*). The remaining non-intersecting and non-overlapping variants were checked for their coverage by projecting the reference positions into the positional framework of the graph and overlapping them with the coverage bed file of the specific accession, using *bedtools intersect*.
The individual call sets were merged into a joined set using *vcf-merge -c any*. The variant length, and allele frequencies were calculated using *vcftools –freq* and *–hist-indel-len*. The full set was then post-processed and two additional sets were created. The first contained only SNPs (*vcftools –remove-indels*), the second had all multiallelic variants split using *bcftools norm -m-any*. The variant allele sizes were categorized as described in subsection 2.2.2. I intersected the SNPs with the highly diverged regions, detected in the *SyRI* analysis, as well as the features in the *araport11* reference annotation (Cheng *et al.*, 2017) using *bedtools intersect*.
In order to compare the level of heterozygosity of the calls made from the graph with a study performed on the linear reference genome (Jaegle *et al.*, 2021), I post-processed them in a similar way. I first subsetted the SNPs to those that intersect with genes from the *araport11* reference annotation and then removed all SNPs with a frequency below 5% of the population (*vcftools –bed araport.genes.bed –max-missing-count 42*).

Chapter 3

Results

3.1 Generation & annotation of new genome assemblies

Six reference genomes were assembled using long-read sequencing technologies, which had finally matured at the onset of my work. Additional methods were used to increase the continuity of the assemblies and their resolution.

3.1.1 Genome assemblies

The assembly of six diverse *A. thaliana* accessions aims at adding additional sequence to the known sequence space. The accessions were chosen to better represent the genetic diversity of the species. As I did not produce the assemblies myself, I will focus on the description on the final chromosome scale assembly and omit the intermediate assembly steps.

Table 3.1: Assembly statistics - Assembly statistics of the six *A. thaliana* accessions. The statistics are presented for the complete assembly, including unplaced contigs, and the scaffolded chromosomes only.

	Complete assembly				Scaffolded chromosomes			
ID	Contigs	Size [Mb]	N50 [Mb]	% of Ns	Size [Mb]	% scaffolded	N50 [Mb]	% of Ns
AT1741	52	121.5	23.3	0.82	118.9	97.96	23.3	0.84
AT5784	66	123.6	23.8	0.92	119.8	96.94	23.8	0.87
AT6909	45	122	23.4	1.08	119.8	98.15	23.4	1.1
AT6911	64	121.6	23.6	0.85	118.6	97.59	23.6	0.87
AT7186	93	125	23.9	2.31	120.7	96.56	23.9	2.39
AT7213	70	122.3	22.8	0.58	117.8	96.36	22.8	0.6

Each scaffolded assembly consisted of 5 long contigs, which represent full scale chromosome sequences with a gap at the core centromeres, with an additional set of unplaced contigs, mostly repetitive sequences from rDNA arrays and centromeres as well as contigs from organellar genomes. The number of unplaced contigs ranged from 40 in *AT6909* to 88 in *AT7186*. The mean number of unplaced contigs was 60. On average 97.3% of the initially assembled contigs had been scaffolded. The accession with the highest contig placement rate was *AT6909* (98.2%), while *AT7213* had the lowest

fraction of scaffolded sequence with 96.4%. The chromosome assemblies contained an average of 1.1% Ns. These had been inserted in gaps in the assembly, based on gap size, estimated using optical maps. This number was largely driven by *AT7186*, which contained 2.4% of Ns. The accession with the lowest amount of Ns was *AT7213* (0.6%). Most Ns were located in consecutive stretches in pericentromeric regions. While not being complete telomere-to-telomere ("T2T") assemblies, a subset of the chromosomes had full-length chromosome arms assembled (Figure 3.4 (B)). The sequence size of each assembly including the unplaced contigs exceeded the size of the current reference genome. The reference had a size of 119.7 Mb, the smallest assembly a size of 121.5 Mb (*AT1741*), while the largest had a size of 125 Mb (*AT7186*). This does not persist when focusing on the 5 chromosomes only. Here the reference had a size of 119.2 Mb. While the mean chromosome assembly size of the six accessions was slightly higher than that (119.3 Mb) three of the assemblies had a shorter chromosome assembly size (*AT1741*, *AT6911*, *AT7213*). The largest was again *AT7186* with a size of 120.7 Mb (Table 3.1).
The dot plots showed that the *de-novo* assemblies were highly syntenic with the *TAIR10* reference genome. Breaks in the synteny were observed in the pericentromeric regions, where additional, or diverse sequences broke the alignment. This was very obvious in chromosomes 1, and 3. In addition a number of large inversions were observed, some of which had been described in previous studies. An inversion on chromosome 4 was present in four out of six genomes, and an inversion on chromosome 5 in all, except for *AT6909*. No large translocations could be observed in this analysis (Figure B.2).

Table 3.2: Assembly *BUSCO* results - *BUSCO* completeness of the six *de-novo* assembled genomes and the *TAIR10* reference genome. *BUSCO* was run using the *embryophyta_odb10* database.

BUSCOs	*AT1741*	*AT5784*	*AT6909*	*AT6911*	*AT7186*	*AT7213*	*TAIR10*
Complete	99.2	99.3	99.4	99.4	99.3	99.4	99.3
- single-copy	98.5	98.4	98.6	98.5	98.5	98.5	98.6
- duplicated	0.7	0.9	0.8	0.9	0.8	0.9	0.7
Fragmented	0.2	0.2	0.2	0.2	0.2	0.2	0.2
Missing	0.6	0.5	0.4	0.4	0.5	0.4	0.5

I ran *BUSCO* with the *embryophyta_odb10* database to assess the completeness of the assemblies. The database contained 1614 *BUSCO* genes. On average 99.3% of them were present as complete copies in the assemblies. Of them 98.5% as single copies. Between 10 genes (*AT1741*) and 7 genes (*AT6909* & *AT6911*) were missing from the assemblies. This result was very similar to the completeness of the *TAIR10* reference genome (Berardini *et al.*, 2015). The only noticeable deviation from the mean of the six *de-novo* assemblies was the percentage of duplicated *BUSCOs*, which was slightly lower in the reference assembly (Table 3.2).

3.1.2 SV calling and comparison to the *TAIR10* reference genome

Comparing novel genome assemblies to an established reference genome is an effective means of detecting structural variants and novel sequences. Here I compare the six *de-novo* assembled *A. thaliana* accessions with the existing *TAIR10* reference genome (Berardini *et al.*, 2015) and report the observed structural variants and novel sequences. I used two different reference based methods to call variants. The first one being the pairwise whole genome alignment based software *SyRI* (Goel *et al.*, 2019). The second one is *vg deconstruct* Garrison *et al.* (2018), a tool to translate variants from a genome graph into a reference space (As described in section 3.2). In addition I subsetted the 1001 Genomes Project short-read calls (1001 Genomes Consortium, 2016) to the six assembled accessions for comparison. I split the detected sequence variation into three different groups, based on their size. Variants where a single base pair was replaced by another single base pair were classified as *SNPs*. In case one allele was equal to or larger than 50 bp the variant was classified as a *large variant*. Everything in between was classified as a *small variant*. Here I describe the results of each method, as well as their overlap and differences.

Short-read based variant calls

The 1001 Genomes Project was a conventional re-sequencing project that aligned short-reads against a linear reference genome (1001 Genomes Consortium, 2016). Thus the length of the called variation was limited and it has been mostly oblivious of structural variation. For the six accessions examined the 1001 Genomes Project had discovered a total of 1,570,148 variants. The majority of them being *SNPs* (92%). *Large variants* had not been called by the 1001 Genome Project. The variants affect a total of 1.7 Mb of the reference genome sequence (Table B.1). 94% of the *small variants* were classified as PAV events. When interpreted in the coordinate system of the reference, 54,876 events were deletion events, affecting 86 kb of the reference sequence. Slightly more variants were classified as insertion events (63,826), adding an additional 96 kb to the sequence.

Table 3.3: Short-read based variant calls - Reference based variant calls of the sixRef accessions made by the 1001 Genomes Project. Variants were classified as *SNP*: if both alleles had a size of 1 bp, as *small variants*: if the largest allele was smaller than 50 bp, or as *large variants*: in any other case.

	AT1741	*AT5784*	*AT6909*	*AT6911*	*AT7186*	*AT7213*
SNPs	453765	357267	523	670840	501009	519305
small variants	29825	19264	562	58413	30667	34367
large variants	-	-	-	-	-	-
total number	483590	376531	1085	729253	531676	553672

On average 445,968 variants were called in each accession. The reference accession *AT6909* contained only 1,085 variants compared to the reference genome and defied the trend of containing mostly *SNPs*. Here the variants were almost equally distributed between *SNPs* (523 variants) and *small variants* (562 variants), with slightly more *small variants* being called. The highest number of variants was discovered in the accession *AT6911* (729,523 variants) (Table 3.3).

Pairwise alignment based variant calls

SyRI (Goel *et al.*, 2019) calls variants from pairwise whole chromosome alignments. As it uses pairwise chromosome allocations it can only call variants within the same chromosome. Variants detected by *SyRI* were divided into two groups. *Structural variants*, such as inversions, translocations, duplications, unaligned sequence, or synteny blocks, and *sequence variation*, that are contained in larger *structural variants*. The *structural variants* were split into two subgroups, rearranged reference sequence and novel sequence. In addition to the description of large structural variants, this section will also zoom in and describe the differences in sequence variation frequency between syntenic and non-syntenic regions of the genomes.

Structural variants:
The largest sub-group of *structural variants* were syntenic regions. On average 9% of the chromosome sequences were marked as being syntenic. 92 Mb of syntenic sequence was shared between all pairwise genome comparisons, covering 77.2% of the *TAIR10* reference genome (Berardini *et al.*, 2015). As it was closest to the reference accession, *AT6909* had the highest amount of syntenic sequence (98.9%) and *AT7213* the least amount (91.4%). The second largest set was composed of unaligned reference sequences that account for 4.1%. The least unaligned reference sequence was found in the comparison with *AT6909* (0.7%), while again the most reference sequence remained unaligned in the comparison with *AT7213* (5.4%). While in the other five accessions inversions (1.8%) accounted for the third largest set of sequences, followed by translocations (0.7%) and duplications (0.3%), in *AT6909* this order was different. Here translocations were the third largest group (0.218%), closely followed by duplications (0.217%) and inversions (0.17%) (Figure 3.1 (A)). In addition to structural variants anchored to the reference sequence, *SyRI* also reported additional, unaligned, sequences in the query genome. These were split into unaligned novel sequences and unaligned duplication-gains. In four out of six genome comparisons the query genome contained less novel than the reference had unaligned sequence. Only *AT5784* and *AT6909* contained more unaligned sequences. The amount of novel sequence varied from 1.1% in *AT6909* to 4.5% in *AT5784*, with an average of 3.6% of the assembled chromosome length. The amount of additional duplicated sequence covered on average 0.8% of the assembled chromosomes and varied between 0.6% in *AT6909* and 0.9% in *AT7186* (Figure 3.1 (B)). Most of the variable sequence was located around the centromere, while the syntenic parts covered most of

the chromosome arms. The five non-reference accessions contained a large inversion near the centromere of chromosome 5 and four of the accessions an additional large inversion on chromosome 4 (*AT5784*, *AT6911*, *AT7186*, *AT7213*) (Figure B.3). Whether the idiosyncratic aspects of some of the assemblies were due to biology or more likely a technical artifacts is not known.

Sequence variation:
The accessions contained between 14,682 (*AT6909*) and 926,496 (*AT1741*) *sequence variants*, with an average of 711,362. The vast majority of them were *SNPs*, followed by *small variants* and a small fraction of *large variants* (Table 3.4) (Figure 3.1 (A)). In addition to the classification by size they were also classified by variant type. Here the larger variants were classified by *SyRI* as PAVs (formerly known as InDels), Copy gain / Copy loss, and Highly divergent regions (HDR). On average each genome comparison with the reference produced 461,909 SNPs. *AT6909* contained the fewest *SNPs* (9,841) and *AT1741* the most (788,570). The second most common sequence category was annotated as highly divergent regions and covered on average 4.9 Mb, in 4,364 events, per comparison. Again *AT6909* contained the least amount of HDR sequence (0.4 Mb kb in 51 events). *AT5784* contained the most HDR sequence, covering a total of 6.0 Mb, distributed over 5,223 events. PAVs accounted for the next largest class, covering for an average of 4 Mb in 151,536 events. From a reference centric view they can be split into insertion and deletion events. While they each cover a similar amount of sequence (Insertion: 1.9 Mb; Deletion: 1.8 Mb), there were twice as many insertion events (101,831), than deletion events (49,704). This trend was identical in four of the genome comparisons. In *AT6909 SyRI* not only called the lowest number of variants, but the relative difference between the covered sequence was also the largest. 58 kb were annotated as deleted sequence (relative to *TAIR10*), while 175 kb were annotated as inserted. In contrast to all other accessions *AT6911* contained slightly more deleted than inserted sequence (2.48 Mb vs 2.37 Mb). All comparisons showed more copy losses of the query sequence than copy gains, compared to the reference genome. In each accession between 248 kb, in 9 events, (*AT6909*) and 660 kb, in 105 events, were annotated as copy loss (avg. 476 kb in 87 events). The sequence variation was not equally distributed in the gen-

Table 3.4: Pairwise alignment based variant calls - Variant calls performed by *SyRI* on pairwise alignments of the *TAIR10* reference genome and the individual genome assemblies. Variants were classified as *SNP*: if both alleles had a size of 1 bp, as *small variants*: if the largest allele was smaller than 50 bp, or as *large variants*: in any other case.

	AT1741	*AT5784*	*AT6909*	*AT6911*	*AT7186*	*AT7213*
SNPs	789297	639545	9821	740918	589490	584798
small variants	133497	228310	4676	175997	162234	188754
large variants	3702	4196	185	4753	4089	3910
total number	926496	872051	14682	921668	755813	777462

ome. I observed a difference between syntenic and non-syntenic regions of the reference. Syntenic regions contained less sequence variation than non-syntenic regions. Over all types of sequence variation, syntenic regions contained on average 63.9 base pairs of variable sequence per kilo base. This more than doubled in non-syntenic sequence, to 158.4 bp/kb. Syntenic regions contained on average 4.8 SNPs per kilo base of sequence, while non-syntenic regions contained 7 SNPs. The amount of PAVs increased from 15.2 bp/kb to 18.2 bp/kb in non-syntenic regions. Highly divergent regions accounted for the most variable sequence. Due to the reference centric view copy gain events accounted for just 0.4 bp/kb in syntenic and 1.2 bp/kb in non syntenic regions. Copy loss variants saw the largest increase in non-syntenic regions. They rise from 4 to 27 bp/kb. This increase in the average base pair count was largely driven by *AT6909*. While all accessions exhibited the same pattern of variant distribution, *AT6909*, again, contained by far the least variation in syntenic regions (5 bp/kb), and the most in non-syntenic regions (176.2 bp/kb). The amount of SNPs and PAVs in non-syntenic regions of *AT6909* were below average, while the amount of copy gains and losses were increased compared to the other accessions (Figure 3.1 (C)).

Graph-based variant calls

*vg deconstruc*t (Garrison *et al.*, 2018) explains variation stored in a genome graph based on a set of reference paths. I used the graph constructed from all six *de-novo* assemblies and the *TAIR10* reference genome (Berardini *et al.*, 2015), as detailed in section 3.2. The resulting variants were split into the previously described three sub groups, based on their size.

Table 3.5: Graph based variant calls - Reference based variant calls extracted from the genome graph by *vg deconstruct*. Variants were classified as *SNP*: if both alleles had a size of 1 bp, as *small variants*: if the largest allele was smaller than 50 bp, or as *large variants*: in any other case.

	AT1741	*AT5784*	*AT6909*	*AT6911*	*AT7186*	*AT7213*
SNPs	425345	499470	9587	603624	466209	455672
small variants	160308	181507	25251	211694	171779	169244
large variants	13127	14282	1288	16215	14023	13794
total number	598780	695259	36126	831533	652011	638710

I called a total of 1,869,605 joined variants from the graph. Of these, 1,272,633 were categorized as *SNPs*. The remaining variants were separated into 539,644 *small variants* and 57,328 *large variants*. The *small variants* had an average size of 4 bp, while the *large variants* were considerably bigger with an average size of 1.5 kb. (Table B.1). 281,558 (47.2%) of the *small* and *large variants* were identified as PAVs. Using the polarity introduced by the reference genome I classified 139,568 of them as insertion type events, adding a total of 12 Mb potentially new sequences to the reference. 141,990 were

classified as deletion type events. They affected a total of 5 Mb of the reference genome. The stark difference in affected bases was, most likely, an artifact of the nested variation and representation in a linear reference based format, and will be discussed later.
The number of detected variants ranged from 36,126 in *AT6909* to 831,533 in *AT6911*. On average I detected 575,403 variable sites per accession using the graph based variant detection approach. *SNPs* were the most common type of variant in all accessions, except *AT6909*, again underlining its special status as an reassembly of the reference accession. In *AT6909 small variants* were more common than *SNPs* and *large variants* combined. In the remaining five accessions *small variants* were the second most prevalent variant category with on average 153,297 variants per accession. The least amount of variants were classified as *large variants* (avg. 12,121) (Table 3.5).

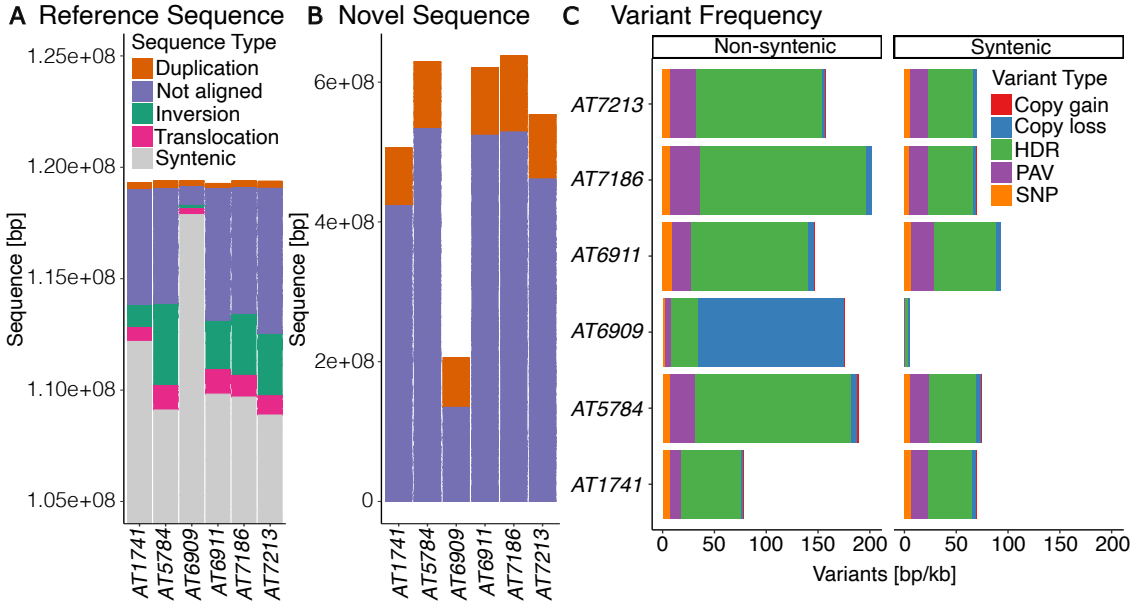

Figure 3.1: *SyRI* variation - Structural variants detected by *SyRI*. Split into Sequence that occurs in the **(A)** reference space and **(B)** novel sequence. **(C)** The frequency of categorized sequence variation in non-syntenic and syntenic parts of the assemblies.

3.1.3 SV set comparison

Genetic variation can be detected by different methods. Here I compare the variant calls of the three previously described reference based methods. The most established method is short-read based variant detection based on the differences of a read pileup to a linear reference genome, represented by the variant set detected by the 1001 Genomes consortium (1001 Genomes Consortium, 2016). The second method, *SyRI* (Goel *et al.*, 2019), calls variants by comparing two full scale assemblies. This allows the detection of longer variants. The third method uses *vg deconstruct* (Garrison *et al.*, 2018) to describe the

variants present in a genome graph derived from a multiple whole genome alignment.
In a first comparison of the number of variants, and their type, called by each of the methods, in each of the accessions I observed a general trend in all accessions, except *AT6909*. The most variants were found in the pairwise whole genome alignments, followed by the graph. The least amount were present in the short-read based sets. In *AT6909* the graph based variant calling method detected more variants than the chromosome based whole genome alignment of *SyRI*. Overall it was the pronounced outlier in the data set. By far the least amount of variation was detected by any of the three methods in this accession. Even the distribution of variant sizes was very different from the other five accessions. While the overall trend in the 5 other accessions was identical, the markedness of it varied. For *AT6911*, *AT7186*, and *AT7213* the difference between the individual call sets was very similar. In contrast *AT5784* contained very little short-read based variants and *AT1741* an overabundance of WGA based variant calls. In all accessions and all call sets, except *AT6909*, *SNPs* were the most abundant variant type (¿75% of the calls), followed by *small variants*, smaller than 50 bp. None of the short-read call sets contained *large variants* ($\geq$ 50 bp). In the two assembly based sets they were the least common type, being slightly more abundant in the graph based variant calls. In the accession *AT6909* the whole genome alignment based variant calls behaved similar to all other sets and accessions, while the graph based method detected more *small variants* than *SNPs* (Figure 3.2 (A)).
For the further analysis I combined the individual calls of each method into non-overlapping call sets. The combined set of the short-read based variant calls contained 1,570,148 unique variants, 92% of them were classified as *SNPs*. The rest were *small variants*. No *large variants* were called from short-reads. The pairwise whole genome alignment calls, by *SyRI*, contained the most unique calls (2,315,844 variants). 78.9% of them were *SNPs* and 28.9% *small variants*. Compared to the other two call sets, this was the highest number and fraction of *SNPs* and *small variants*. The remaining 0.3% were *large variants*. The combined graph based variant calls summed up to 1,869,605. 68% of them were classified as *SNPs*. This was the lowest number and fraction of *SNPs* in the three sets. 28.9% were *small variants* and 3.1% were *large variants*, the highest number of all three combined call sets (Figure 3.2 (B)). While the whole genome alignment derived variant set contained more overall variation, the amount of affected bases was lower than that of the graph based variants. Here the large variants affected a total of 85.2 Mb (Figure 3.2 (C)). The larger size of graph derived variation was also visible in the distribution of variant sizes. Here the graph-derived variation was consistently more prevalent in larger variants (Figure 3.2 (D).
Merged together, the three variant sets contained 3,377,162 unique variants. 28% of them were shared between all three sets, 22.5% are shared between two of the methods, and the remaining 49.5% of the variants are private to one of the methods. The highest fraction of intersecting variants were found in the short-read based calls (60.2%), followed by the graph based variants with 50.6%. Only 36.5% of the whole genome alignment based *SyRI* calls were shared with the other two methods. The variants detected by *SyRI*

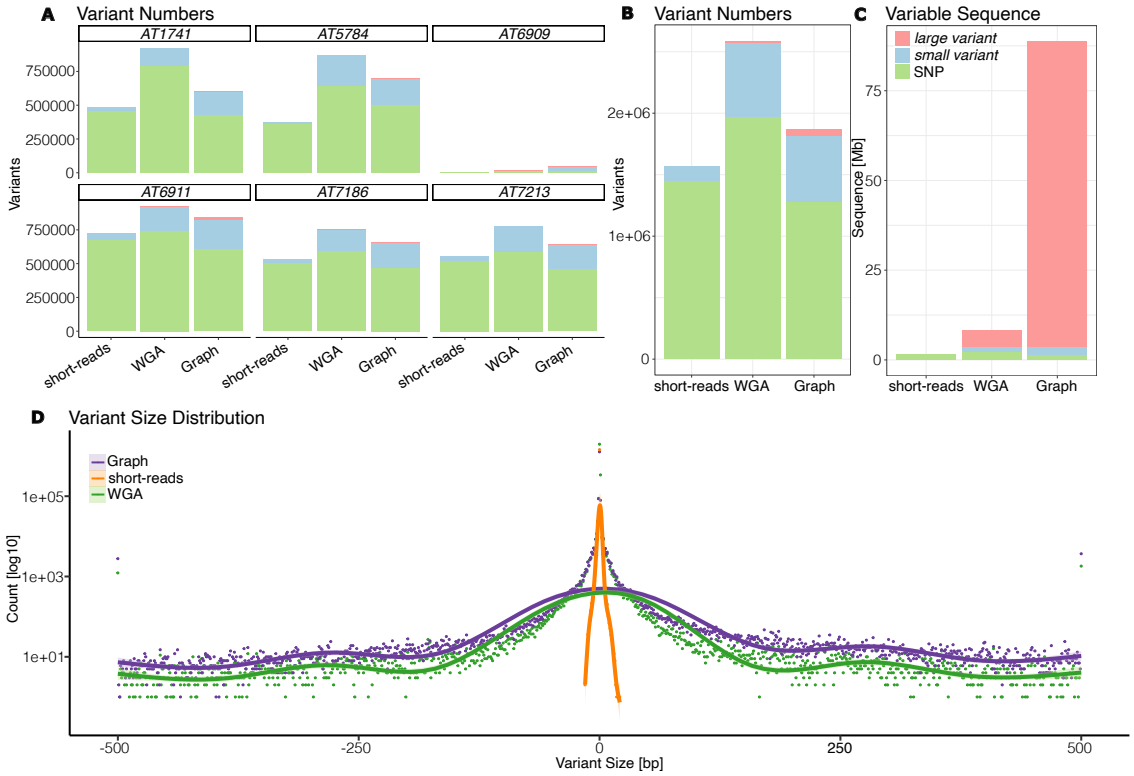

Figure 3.2: Variant comparison - **(A)** Number and type of variants in comparison to the *TAIR10* reference genome, detected by the three methods from six de-novo assembled accessions of *A. thaliana*. **(B)** Number of combined variants detected by each of the methods over all accessions. **(C)** Amount of sequence space found to be affected by variation. **(D)** Distribution of variant sizes in the three sets. Variants larger than 500bp have been combined into one category.

were private in 37.1% of the cases, the highest fraction and count. The fraction of private graph based variants was 31%, and the short-read derived variants were private in 8.3% of the cases (Figure 3.3 (A)).
In the individual size categories, the pattern was similar, but overall *SNPs* were shared more often than *small variants* or *large variants*. 70.6% of all *SNPs* detected in the graph were shared with the other two methods, in the short-read based SNP set 62.3% were shared. Only 45.8% of the *SNPs* called from the pairwise whole genome alignments were shared with the other two methods. *SyRI* called the lowest number of shared *SNPs* and the highest number of private ones (26.7%). Just 10% of the *SNPs* detected in the graph were private, and the short-read based calls contained only 7.2% of private *SNPs* ((Figure 3.3 (B)). *Small* and *large variants* were shared less frequently. 36.9% of the *small variants* detected in the short-read data were shared with the other two call sets, while 21.1% were private. The largest fraction, 37.7%, were shared with *SyRI*. Only 7% of the *small variants* called by *SyRI* were shared with both of the other methods. 69.9% of the *small variants* detected by *SyRi* were private. While the short-read variant

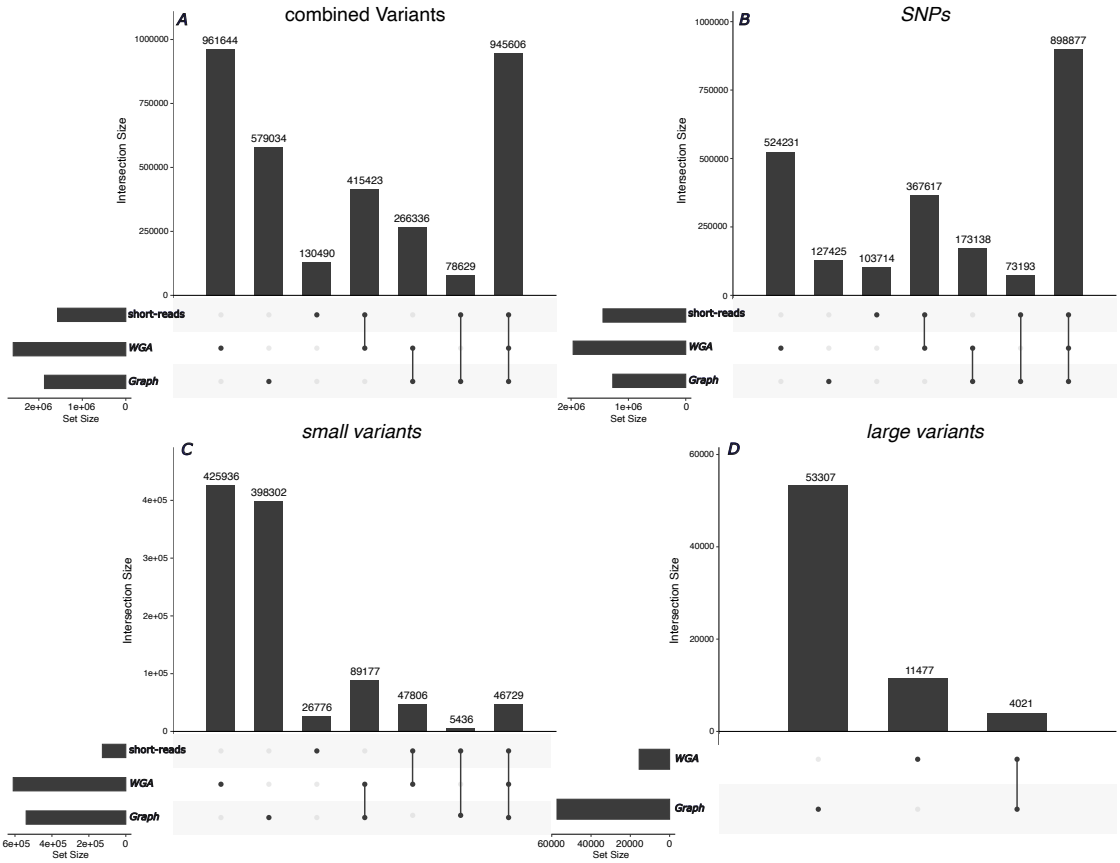

Figure 3.3: Variant intersection - Intersection of variants detected by the three different methods. The variants were intersected as the full set of **(A)** all combined variants, **(B)** *SNPs*, **(C)** *small variants*, and **(D)** *large variants*.

calls had the largest intersection with *SyRI*, this intersection accounted for only 7.8% of the *small variants* detected by *SyRI*. The largest pairwise intersection of *small variants*, detected by *SyRI*, was with the graph (14.6%). For the graph based calls 8.7% of the *small variants* were shared with the other two methods, and 73.8% were private. Again, the largest pairwise intersection was observed with the pairwise alignment derived variants (16.5%) ((Figure 3.3 (C)). No *large variants* were detected by the short-read based method, this leaves just the variants detected by *SyRI* from pairwise alignments and variants called from the graph to compare. As *SyRI* overall called fewer variants, a larger fraction was shared with the graph calls (25.9%). Just 7% of the large variants detected from the graph were shared with those called by *SyRI*. The remaining variants were private to their method ((Figure 3.3 (D)). The reason for the overabundance of *large variants* called from the graph will be discussed later.

The *small* and *large variants* that were not shared among all variant sets were subjected to a positional overlap with *SNPs*, *small variants*, and *large variants*, to detect fuzzy alignment borders or variants that were resolved to different levels of precision. Despite

the higher number of variants called from the pairwise alignments, only 12.3% of the *small variants* detected from the short-read alignments overlapped with the calls made by *SyRI*. In contrast, 55.4% of the *small variants* were overlapping with the calls made from the graph. On average 1.4 variants detected by *SyRI* were contained in a short-read alignment derived *small variant*, whereas 2.7 variants detected in the graph were overlapped with a *small variant* from the short-read call set. Only 4.8% of the *small variants* detected by *SyRI* overlapped with variants called from short-reads, with an average of 1.3 short-read variants per small *SyRI* derived variant. Again the graph yielded a larger fraction of overlapping *small variants*. Here 19.7% of the *small variants* overlapped with an average of 3.1 graph variants contained in a *SyRI* variant. The difference in the overlap of graph based *small variants* was less pronounced. 16% of the *small variants* overlapped with an average of 1.3 short-read based variants, and 22.8% were overlapped with variants called by *SyRI*. Here a graph derived *small variant* contained an average of 1.5 *SyRI* variants.
As *large variants* covered a larger amount of sequence the average number of intersecting variants increased. 31.3% of the *large variants* called by *SyRI* overlapped with variants from the short-read based variant set. Here each variant contained an average of 9.6 smaller variants. The fraction of *large variants* that intersected with variants detected from the graph was bigger (44.5%), but those contained fewer variants (6.2). Fewer *large variants* called from the graph were overlapping with variants from the other two methods. 20% of them overlapped with variants detected by the short-read based method, while 20.6% were overlapping with *SyRI* derived variants. While the fractions were lower than those of *SyRI*, the large variants detected in the graph contained a higher number of smaller variants. On average 24.5 short-read derived variants were contained in a large graph variant. For the pairwise alignment derived variants this number increased to 44.5 shorter variants per large variant from the graph (Table B.2).

3.1.4 Genome annotation

Auto-ant annotation pipeline validation

The *auto-ant* annotation pipeline was designed to improve the annotation quality by combining multiple tools into one framework and merging the annotations in a weighted form. In order to choose the correct weights for each annotation I had to validate their performance. This was done by re-annotation of the existing *TAIR10* reference genome (Berardini *et al.*, 2015) and comparing the results with the *araport11* reference annotation (Cheng *et al.*, 2017). This reference annotation has, in parts, been curated manually and is therefore closer to a truth set, than a purely computational annotation.
The individual annotation tools showed different accuracies for different features of the reference annotation. The numbers differed from tool to tool, and some of the features were not discovered at all by some tools. Features that were annotated in the *araport11* annotation, but were not found in any intersection with the new annotations are: long

non-coding RNAs (lncRNAs), microRNAs (miRNAs), transfer RNAs (tRNAs), and ribosomal RNAs (rRNAs). In the evaluation I mainly focused on the annotation of protein coding sequence (CDS), exons (CDS plus untranslated regions [UTRs]) and full genes. Transcripts assembled by *cufflinks* (Trapnell *et al.*, 2010), from *AT6909* RNA sequencing data covered 61.1% of the CDS entries in the *araport11* annotation, but none of the calls had the same boundaries as the reference calls. *SNAP* (Korf, 2004) detected 72.3% of the reference CDS entries, but again none of the predicted features had the same boundaries. In the RNA supported *augustus* (Stanke *et al.*, 2008) annotation 87.6% of the reference CDS entries were covered by 89.42% of predicted CDS entries. The value slightly dropped for the features predicted with the *BUSCO* (Seppey *et al.*, 2019) retraining. Here 70.5% of the reference CDS entries are covered by 83.2% of the annotated entries. Using the hints produced by *LiftOff* (Shumate and Salzberg, 2020) I was able to detect 88% of the CDS features also present in the reference, corresponding to 87.9% of the features annotated in the new prediction. On the level of individual exons, the *cufflinks* transcript assembly had a sensitivity of 43.9%, detecting 64.8% of the reference features. *SNAP* overlapped with 39.1% of the reference exons, with a specificity of 80%. In the *augustus* predictions the RNA sequence supported annotation detected 62.7% of the reference exon features. The correct features accounted for 49.2% of all features detected. The *BUSCO* retrained annotation intersected with 39% of the reference exons, but the exact exon borders were often imprecise. The *LiftOff* supported annotation had an exon detection sensitivity of 52.3% and a specificity of 67.7%. The imprecise exon borders massively influenced the accuracy of the annotated gene features. For full-length gene entries cufflinks performed poorly. Almost no gene features in the reference annotation were fully covered. This was a result of the different definition of the gene feature in the reference annotation that also contains three, and five prime UTRs. For *SNAP* the sensitivity was the highest among the individual gene predictions (1.8%). The second RNA supported prediction, using *augustus*, intersected with 0.9% of the full length reference genes, with a specificity of 0.1%. The *BUSCO* retrained run of *augustus* had a specificity of 1.4%, and the by far best sensitivity of 55.9%. The lift over supported *augustus* prediction had a perfect match sensitivity of 1.3% and a specificity of 0.4% (Table 3.6).

Table 3.6: *auto-ant* validation - Sensitivity and specificity of the annotation methods, and the combined annotation result. The sensitivity and specificity were calculated by intersecting the annotated features with the *araport11* annotation. Sensitivity is the fraction of *de-novo* annotated features being fully contained in features of the reference annotation. Specificity is the fraction of correctly annotated reference features.

	Annotation feature	*Augustus* runs *BUSCO* retraining	*LiftOff* evidence	RNA-Seq evidence	*SNAP*	*cufflinks* assemblies	*EvidenceModeler* results
Sensitivity	CDS	70.5	88	87.6	72.3	61.1	86.7
Specificity	CDS	83.2	87.9	89.4	0	0	91.7
Sensitivity	exon	39	52.3	49.3	39.1	43.9	46.1
Specificity	exon	0	67.6	62.7	80	64.8	91.7
Sensitivity	gene	1.4	1.3	0.9	1.8	0.01	2
Specificity	gene	55.9	0.4	0.1	0	0	78.3

Table 3.7: *evidenceModeler* weights - Weights used in the *evidenceModeler* processing of the individual annotations. The weights were assigned based on the trustworthiness of the individual annotations. Annotations with RNA support were rated higher. In addition the weights were chosen to balance the bias of the three *augustus* runs. The *ab-initio* weights were assigned based on the same analysis.

	Evidence weights	
	RNA supported annotation	full ab-initio annotation
augustus - *BUSCO* retraining	2	2
augustus - *LiftOff* evidence	3	3
augustus - RNA-Seq evidence	3	-
SNAP	4	2
cufflinks assembies	5	-

The combination of the five different annotation methods into a joint annotation with *evidenceModeler* (Haas *et al.*, 2008) required a weight matrix that rates the trustworthiness of the input data. I created this matrix based on the re-annotation performance of the tools and gave extra weight to the RNA-evidence based methods compared to the purely similarity-based ab-initio predictions. In addition the weights were set in a way that if two non-*augustus* annotations disagreed with *augustus*, they could overrule the three *augustus* based annotations. This was done to minimize the bias introduced by the three independent *augustus* runs.

This resulted in a weight matrix where *cufflinks* results were given the highest weight, 5, followed by *SNAP* with a weight of 4. The RNA-based, and the *LiftOff* based *augustus* runs were both assigned a weight of 3. The run of *augustus* that used a retrained prediction model from *BUSCO* was rated lowest with a weight of 2. In addition I created a weight matrix that could be used without supporting RNA reads. Here the weights were set in a way that two out of the three remaining tools were able to overrule the remaining tool. The highest weight of 3 was attributed to the *LiftOff* based *augustus* annotation. The weight of the remaining two tools was set to 2 each (Table 3.7).

The weight matrix was used for *evidenceModeler* with the gene predictions described above. It was able to detect 86.7% of the reference CDS features, which was slightly less than the individual annotations of the RNA, or *LiftOff* supported *augustus* annotations,

but the specificity increased to 91.7%. The same was true for the exon features. Here the sensitivity droped slightly below the highest individual annotations (46.1%), but the specificity greatly increased to 91.7%. While the specificity for full gene features was very low, at 2.0%, it was already higher than the individual annotations. The specificity of the combined and weighted calls was at 78.3%, much higher than the individual assemblies. The sensitivity of full gene features remained low at 4.9%. The specificity, however, increased to 75.6% (Table 3.6). The performance of the combined annotation on some features that were not used in the creation of the weight matrix was also notable. Only very few pseudogenes present in the reference annotation were found to intersect with the *de-novo* predictions. The same was true for TEs (Table B.3).

Six new reference genomes

The comparison of sequence features between multiple genomes allows for a better understanding of their evolutionary history and enables us to discover footprints of past genetic events in the genome. Therefore I annotated different types of features in the six *de-novo* assembled *A. thaliana* genomes, including repeats and TEs as well as genes.

TE and repeat annotation

TEs and repeats were detected independently using *EDTA* (Ou *et al.*, 2019) and *repeatMasker* (Smit, AFA, Hubley, R & Green, P., 2013). On average 12% of the assemblies were classified as TEs that do not intersect with annotated genes. The number of detected TEs ranged from 24,183 in *AT6909* to 27,612 in *AT1741* (mean: 25,659). Almost half of the annotated TEs were classified as Helitrons (mean: 43.82%). *AT6911* contained the highest number (12,571), while *AT6909* contained the fewest helitrons (10,572). The second largest category were Gypsy LTR retrotransposons. On average they accounted for 20.85% of the annotated features. This time *AT6911* contained the least of them (4,882) and *AT1741* contained the most (6,390). All other categories contributed less than 10% each. The smallest group, Polintons, were not found in every assembly. Only *AT1741* (70), *AT6909* (65), and *AT7213* (64) contained any. On average, 7% of the assembled sequence was classified as repeats by the merged results of *repeatMasker* and *EDTA*. *AT7186* contained the highest fraction of repeat sequence (7.7%) and *AT1741* the lowest fraction (6.6%) (Figure 3.4 (C)).

Gene annotation

Gene annotations provide an insight into the metabolic and adaptational arsenal available to an individual accession. In this section I describe the gene annotations of the six *de-novo* assembled *A. thaliana* genomes, and compare them with the existing *araport11* reference annotation (Cheng *et al.*, 2017), the annotation of the outgroup, *A. arenosa*, and with each other. I also describe the observed changes in order and orientation of

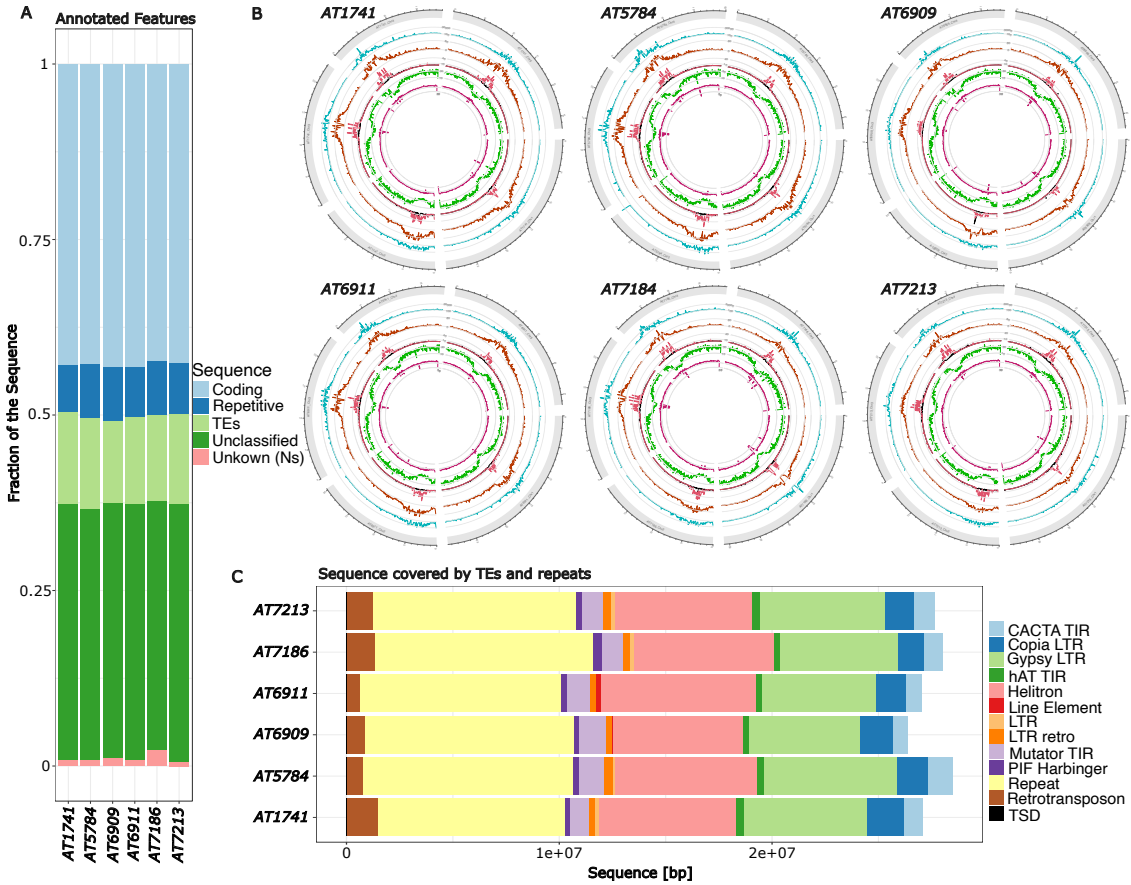

Figure 3.4: Genome annotation - **(A)** Fraction of the assembled sequence covered by different features. **(B)** Circos plots of the six *de-novo* assemblies to show the distribution of features along the chromosomes. Description of tracks from the outside inwards: Repeats (blue); TEs (red); Copia (pink) and gypsy TEs (black); genes (green); GC content (pink). A stark depression in this track is a result of assembly breakpoints. **(C)** Amount of sequence covered by the different types of TEs and repeats in each of the six assemblies.

orthologous genes along the genomes. Finally the six assembled genomes enable me to attempt to describe parts of the *A. thaliana* proteome from a reference perspective as well as from an unbiased pan-proteome perspective. By doing so I explore the expanding and contracting orthogroups as well as the conserved core proteome.
Genes in the *de-novo* assembled genomes were annotated and assigned to orthogroups using the *auto-ant* pipeline subsection 2.1.1. Genes were classified into three groups based on the orthogroup assignment. They were classified as *aralogs* if the gene was orthologous to at least one reference gene, as orthogroup genes (*OGgenes*) if an ortholog of the gene was found in at least one other annotation aside from the reference genome, and as unassigned, *private* genes if no orthologous gene was found in any other annotation of the *de-novo* assembled genomes. In addition, orthogroups were classified as standard and non-standard, based on changes in their copy numbers, or localization in

the synteny. The RNA sequencing reads were mapped using *STAR* (Dobin *et al.*, 2013) and originated from four different tissues. The number of overall reads and per-tissue reads greatly differed between accession. After quality trimming on average 90% of the reads remained. The amount of mappable reads was very low. Only between 39% and 63% of the reads were mapped to the corresponding genome assembly (Table B.4). Using *auto-ant* between 25,644 genes, in *AT1741*, and 26,032 genes, in *AT6909*, were annotated in each genome (mean: 25,827 genes). On average 78.8% of the annotated genes were supported by RNA evidence. The annotated genes were distributed in a pattern that was the reverse of the distribution of the TEs. Genes were less common in the pericentromeric region and more common on the chromosome arms (Figure 3.4 (B)).
In the orthogroup assignment over 95% of the annotated genes were categorized as *aralogs*. *AT6909* carried the highest number of *aralogs*, and *AT6911* the lowest. Only a minority of the genes could not be assigned into orthogroups with *araport11* genes. Each of the six *de-novo* annotations contained a set of roughly 1,300 genes that were classified as *OGgenes*. Less than 140 genes in each of the *de-novo* assemblies were classified as *private*. In addition 1,781 genes from the *araport11* reference annotation were not allocated to an orthogroup and were classified as *private* (Figure 3.5 (A)).
Including *araport11* reference and the *A. arenosa* outgroup, a total of 25,911 orthogroups were constructed. 2,002 orthogroups did not have members from the six *de-novo* annotations. Of these, 53 contained genes from reference and outgroup, 635 only from the outgroup, and 1,314 only from the reference. The remaining 23,909 orthogroups contained at least one *de-novo* annotated transcript. These were classified into 17,166 orthogroups that contained transcripts from both the reference annotation and the outgroup, 4,793 just from the reference, 351 just from the outgroup. 1,599 orthogroups only consisted of transcripts from the newly annotated genes (Figure 3.5 (D)).
The number of accessions contributing to each orthogroup had a U-shaped distribution, typical for pan-genomic datasets. The largest group represented orthogroups with members from all six accessions. Of them 16,488 (84%) also contained members from the outgroup and reference. Only 333 of the orthogroups consisted solely of *OGgenes*.
I also discovered 148 orthogroups that contained at least one ortholog from the six *de-novo* annotated genomes and from the outgroup, but none from the *araport11* reference. The majority of these non-reference orthogroups consisted of one-to-one orthologs, with only 53 of them having one or more accessions contributing multiple members. The total set consisted of 1,632 transcripts of *de-novo* annotated genes from the six assemblies. Of these transcripts, 30.3% were supported by RNA evidence. 62.9% were located inside of variable regions detected by *panSV* in the genome graph (subsection 3.2.3).
On the left side of the pan-transcriptome distribution 872 orthogroups contained transcripts that were only annotated in one of the six *de-novo* annotations. Despite only containing genes from one of the six *de-novo* annotations, 284 of them had at least one ortholog from the reference annotation or the outgroup annotation (Figure 3.5 (B)). The distribution that could be observed for the number of accessions per orthogroup was also observed when comparing the number of genes per orthogroup. Here the U-shaped dis-

tribution was repeated at multiples of eight, the number of annotations (6 *de-novo*, 1 reference, 1 outgroup) that had been used in the orthogroup assignment. With the first block being the largest, most of the orthogroups only contained a single copy from each contributing accession. The next blocks showed a similar U-shaped pattern. The majority of orthogroups, beginning from eight genes per group, were complete orthogroups. This means they contained genes from all eight annotations (Figure 3.5 (C)). To ensure the consistency with the curated reference annotation the single longest transcript of each gene was compared to its reference orthologs. The majority of genes had the same, or a very similar length. Genes with different lengths than their reference counterpart tended to be longer (Figure 3.5 (E)).

A subset of the orthogroups were classified as non-standard orthogroups as they had variable gene copy numbers, or contained translocated gene copies. This group consisted of just 6,724 orthogroups, with a total of 40,266 genes. 4,775 of these orthogroups contained at least one reference gene. In total 7.291 reference genes were found to be part of non-standard orthogroups. A GO-term analysis of the reference genes revealed that pathways that were involved in apoptosis, defense mechanisms and cell-to-cell signaling were enriched in this set. The set of non-standard orthogroup genes will be intersected with structural variation events in subsection 3.2.3 in an effort to describe them further in the pan-genome sequence context.

Considering the different annotations and the fraction of Ns in the assemblies, on average 36.2% of the sequence remained unclassified. This varied between 35.4% in *AT7186* and 36.7% in *AT7213*. The majority of the remaining sequence was annotated as a coding sequence. It ranged from 42.3% in *AT7186* to 43.0% in *AT1741* (mean: 42.9%). Over all six assemblies, 12.6% of the assembled sequence was annotated as TEs. Here *AT6909* contained the least (11.8%) and *AT1741* the most (13.2%). Apart from the fraction of Ns, annotated repeats covered the least sequence in the genome. On average 7.3% of the sequence space was annotated as repetitive. *AT1741* contained the least repetitive sequence (6.6%) while *AT7186* contained the most (7.7%) (Figure 3.4 (A)). The annotated features were distributed along the chromosomes as expected: TEs were enriched around the centromeric regions, fading out along the chromosome arms, with a similar distribution for other types of repeats. Notably copia LTR TEs were more common along the chromosome arms, while gypsy LTR TEs had their highest peak in the centromeric regions. The distribution of genes was inverted compared to those of repeats and TEs. The GC content of the sequence was similar throughout the genome, with obvious drops at the locations where contigs were merged, and separated by Ns in the assemblies (Figure 3.4 (B)).

Pan-proteome exploration

I explored the pan-genome from different angles. First by applying a reference-centric approach of categorizing expansion or collapse of orthogroups based on the copy number of reference genes. Then in a true pan-genome sense by excluding the reference and categorizing orthogroups of the *core* and *shell* genome based on the median copy num-

ber in the orthogroup. As a last step I look at the positional conservation of orthogroups and define a set of interesting non-standard orthogroups for intersection with structural variation calls.

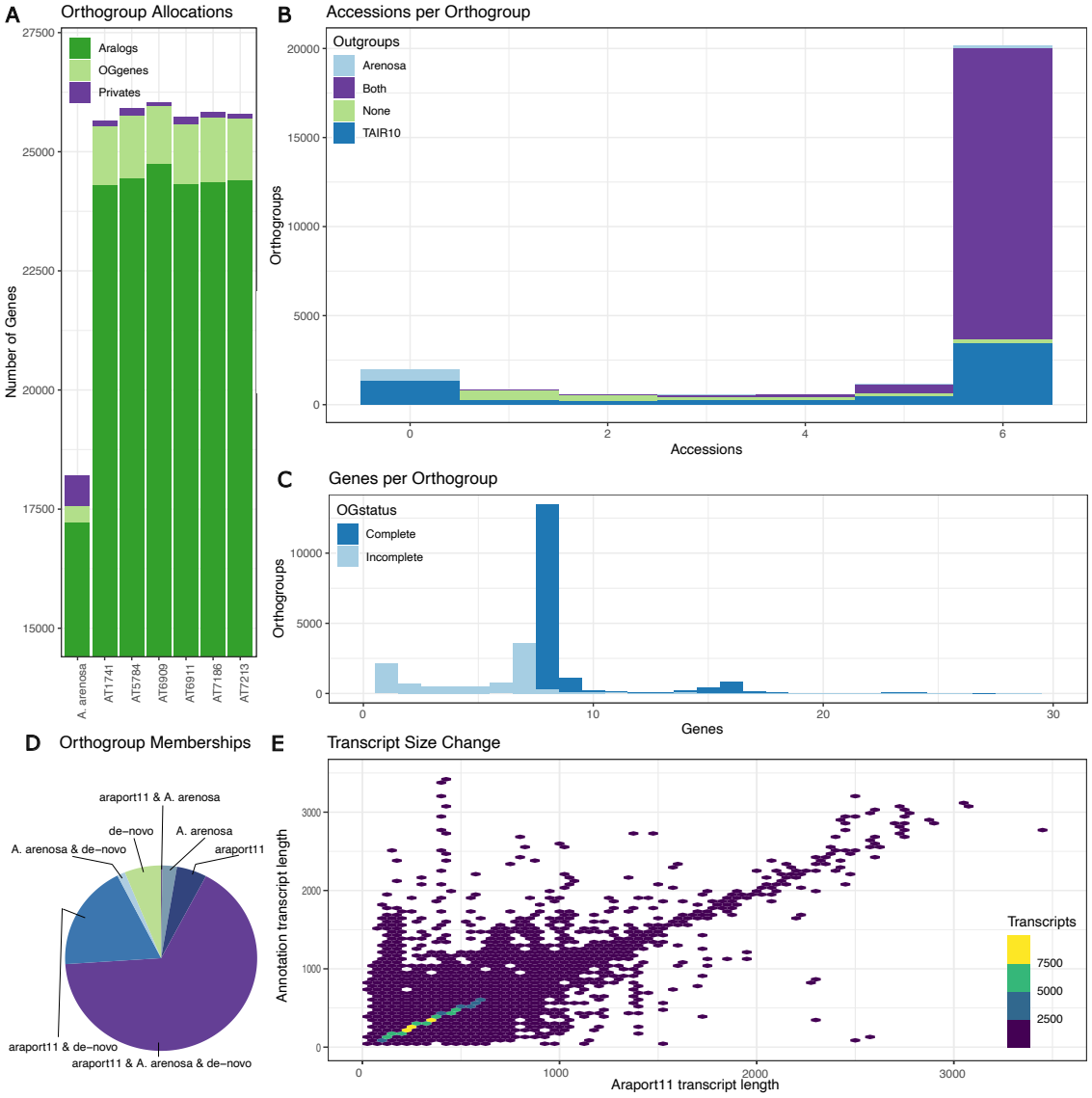

Figure 3.5: Orthogroups - **(A)** Allocation of genes into the three orthogroup types. *Aralog*: if a transcript from the *araport11* reference annotation is part of the orthogroup. *OGgene*: if at least one other accession is contributing to this orthogroup. *Private*: if no other accession is contributing transcripts to the orthogroup (can be single-transcript orthogroups) **(B)** Number of the accessions present in an orthogroup. Colored based on the presence of outgroup, and/or *araport11* reference annotation in the orthogroup. **(C)** Number of genes per orthogroup. An orthogroup is considered to be complete if transcripts from all six *de-novo* assemblies are present. **(D)** Distribution of reference, outgroup, and *de-novo* transcripts in orthogroups. **(E)** Changes in transcript length compared to the length of *araport11* transcripts in the same orthogroup.

In each accession over 90% of the calculated orthogroups had the same copy number as the reference annotation. The expanded orthogroups made up between 0.8%, for *AT6909*, and 1.4%, for *AT6911*, of all orthogroups. For each accession between 2.7% and 3.1% of the orthogroups were contracted. Again *AT6909* had the fewest of these orthogroups and *AT6911* the most. The rest of the orthogroups either did not contain reference transcripts, or were *private* to this accession (Figure 3.6 (A)). In the true pan-genome analysis the 23,909 orthogroups that contained transcripts from the *de-novo* assemblies were first classified into *core* orthogroups that contained at least one transcript from each of the six accessions, or *shell* orthogroups that lacked transcripts from at least one of the assemblies. In total 20,150 (84.3%) orthogroups were classified as *core*, 12.1% as *shell*, and 872 orthogroups (3.6%) contained only transcripts from a single accession (Figure 3.6 (B)).
In each accession the majority of core orthogroups were also classified as *conserved*. This means they had the same number of transcript copies as all other contributors. For those orthogroups that were not considered as *conserved*, more were *expanded*, than *contracted* (Figure 3.6 (D)). The number of *shell* orthogroups varied between 2,107 and 1,866 per accession. Again the majority of them were *conserved* orthogroups. On average, 142 orthogroups only contained genes from a single accession and thus were *private*, an average of 88 of such orthogroups were *expanded* and an average of 51 were found to be *contracted* (Figure 3.6 (E)).
A combined analysis of all variable *shell* and *core* orthogroups showed that orthogroups that were part of the *shell* pan-proteome were more variable in their copy numbers than orthogroups that were part of the *core* pan-proteome. Orthogroups from the *core* pan-proteome had many instances where only a single accession had a different copy number. In most cases this was an increased copy number (Figure 3.6 (G)). A bootstrapped saturation analysis of the pan-proteome exhibited signs of a beginning saturation (Figure 3.6 (C)). The genomic location based analysis revealed a high degree of conservation along the chromosomes for core single-copy orthogroups. The vast majority of them were conserved in the same order and orientation in as in the reference. Only a minority of them were inverted or translocated to a different location in at least one of the accessions (Figure 3.6 (D)).

Figure 3.6: Pan-proteom - **(A)** Reference based contraction and expansion of orthogroups compared to the number of copies in the *araport11* annotation. **(B)** Pan-proteome distribution of orthogroups in the six *de-novo* assemblies **(C)** Bootstrapped saturation analysis of orthogroups. **(D)** Contraction and expansion of *core* orthogroups in the pan-proteome. Compared to the median copy number of the orthogroup **(E)** Contraction and expansion of *shell* orthogroups. **(F)** Synteny plot showing the order and orientation of genes in the assemblies. Genes that remain syntenic are coloured gray. Translocated genes are coloured blue, and inverted genes are coloured red. **(G)** z-Score analysis of copy number changes in orthogroups. Missing accessions were assigned the value -6 to make them clearly visible.

3.2 Graph genome

The graph genome that I have constructed has unique properties and requires novel methods for analysis. In this section I am first going to describe the graph in its different phases of construction and will analyze and validate the portion of the graph that remained unaligned to the *TAIR10* reference genome (Berardini *et al.*, 2015) in an effort to estimate the graph alignment quality. I will then explore the graph based pan-genome. Next I am going to show the performance and results of my novel pan-genome graph based variant detection tool *panSV*, compare and benchmark different algorithms for short-read alignment to a genome graph and genotype the 1001 Genomes short-read data (1001 Genomes Consortium, 2016) using the graph as the alignment reference.

3.2.1 Graph construction

The graph was constructed using the *pggb* pipeline, which combined different tools (Garrison *et al.*, 2023). The initial all-versus-all alignment resulted in 9,627 individual alignment blocks that were converted into 5,021,366 nodes in the first *seqwish* graph, 7,150,481 edges connected the nodes in the graph resulting in an average node degree of 1.42. The sequences in the first graph summed up to 173 Mb. This was a compression to 20.2% of the input sequence lengh. The average node length was 34 bp (median = 13 bp). In the next step the graph was submitted to a sorting and local realignment to resolve previous misalignments and gaps. This resulted in the final graph that was used in all other downstream analysis. In the final graph the number of nodes increased to 6,660,734, connected by 9,113,038 edges. This resulted in a slight drop in node degree (1.37). Due to the realignment the graph length was reduced to 19.7% of the input sequences with a total sequence length of 169 Mb. The realignment reduced the mean node length to 25 bp (median = 1 bp). The majority of the nodes in the graph were traversed only once by each path. 218,670 nodes (3.4%) were considered as repetitive nodes as at least one path traversed them multiple times. With a mean length of 8 bp (median = 1 bp) those nodes were shorter than the non-repetitive ones (mean length = 26 bp; median = 1 bp) (Table 3.8). Each input genomes has been compressed in the graph construction process. On average the length of nodes touched by each accession were 2.9% shorter than the assembly size. *AT7186* had the highest compression (3.6%), while the *TAIR10* reference (Berardini *et al.*, 2015) had the least amount of compressed sequence (2%). The 2D layout of the graph showed a strongly connected center, where almost all subgraphs aligned, with long tendrils and loops stretching out from it (Figure 3.7).

Table 3.8: Graph construction - Basic statistics on the individual graph construction steps.

				Node Sizes	
	# Nodes	# Edge	Mb	Mean	Median
seqwish graph	5,021,366	7,150,481	173	34	13
Final graph	6,660,734	9,113,038	169	25	1

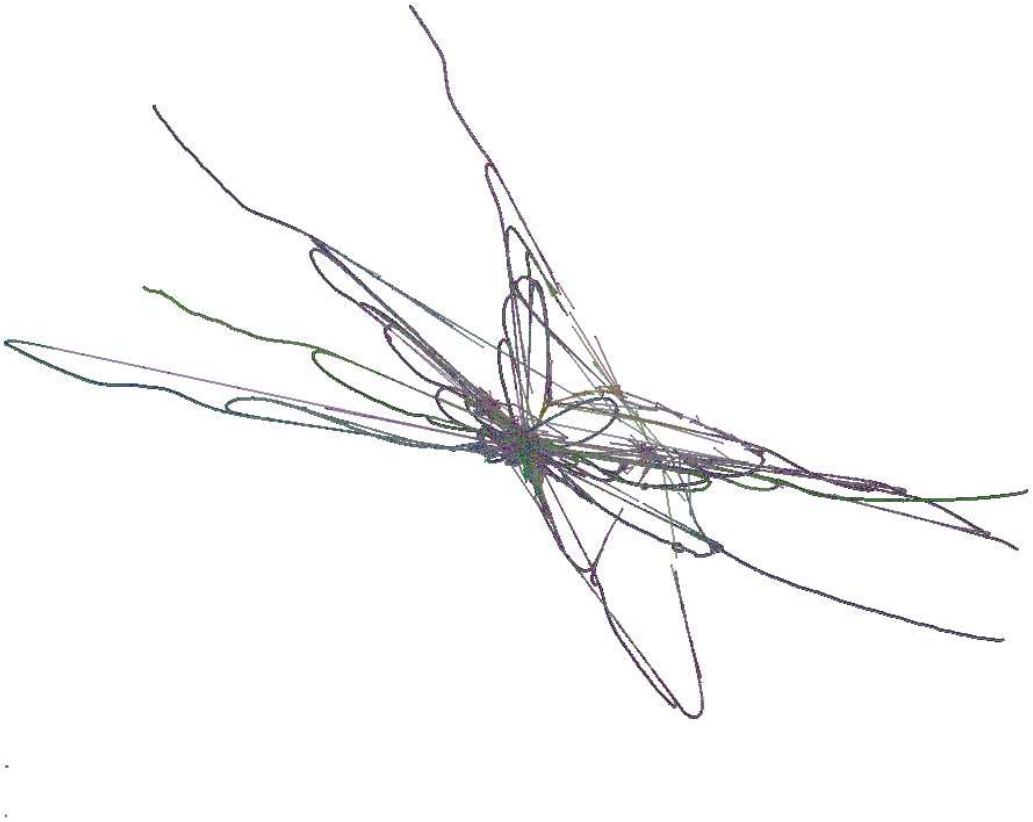

Figure 3.7: 2D Graph Layout - Two-dimensional layout of the graph constructed from the six *de-novo* assembled *A. thaliana* accessions, and the *TAIR10* reference genome. Including all chromosomes and contigs. The pericentromeric regions are drawn together in the center of the layout, while the chromosome arms extend out from there.

3.2.2 Graph pan-genome

The basic statistics of the graph can already tell us something about the relatedness of the accessions within the graph and their relationship with the reference genome. We can further extract information about previously unobserved sequence from the new accessions.

Table 3.9: Graph pan-genome - Pan-genome described by the graph. The fraction always references the value of the full graph of this category.

					Node Sizes	
	Nodes	Percent	Mb	Percent	Mean	Median
Full graph	6694806	-	168.8	-	25.2	1
Core graph	1767213	26.4	92.7	54.9	52.4	18
Shell graph	3525563	52.7	37.3	22.1	10.6	1
Private graph	1402030	20.9	38.8	23	27.6	1

The distribution of accessions per node in the final graph showed a U-shaped distribution, similar to the distribution of the pan-proteome (Figure 3.8 (A)). 52.7% of all nodes belonged to the *shell* genome, which was made up from sequences that were traversed by two to seven genomes. The *core* genome was made up of 26.4%

of the nodes. 20.9% of the nodes were *private*. The values drastically changed when considering the sequence content of the nodes. While only a quarter of the nodes were part of the *core* genome, they contained 55% of the sequence in the graph and were considerably longer than the other pan-genome categories (mean: 52 bp; median: 18 bp). *Private* and *shell* sequence each accounted for 22.5% of the graph's total sequence content (Figure 3.8 (B)). The *private* nodes (mean: 28 bp; median: 1 bp) were on average longer than the *shell* nodes (mean: 11 bp; median: 1 bp) (Table 3.9) (Figure 3.8 (C)). On a per-genome level, 78.1% of each genome sequence was part of the *core* genome, ranging from 79% for the *TAIR10* reference genome (Berardini *et al.*, 2015) to 76.9% in *AT7186*. The *shell* genome covered on average 17.3% of the genomes, ranging from 20% in *TAIR10* to 13.6% in *AT6911*. The two extremes were reversed for the fraction of *private* sequences. Here *AT6911* contained the highest amount (8.1%) and *TAIR10* the lowest amount (0.9%). The average was 4.7% of *private* sequence per genome. As the *TAIR10* and *AT6909* sequences were highly identical, the 0.9% *private* sequence was not a true representation of their private subgenomes. The smallest amount of *private* sequence, outside the two highly similar genomes was found in *AT7213*, with 4.4% (Figure 3.8 (F)). On average 2.9% of each genome was collapsed in the graph representation due to repetitive sequence content. (Table 3.10) In the saturation analysis neither the *core* nor the *shell* genome saturated (Figure 3.8 (D)).

Table 3.10: Path core levels - Per accession graph and pan-genome statistics. 'Comp. Seq.' describes the fraction of the assembled sequence that has been compressed in the graph representation. Repeated nodes are counted just once. The pan-genome percentage is the percentage of sequence in the graph that are traversed by this accession. Path percentage describes the fraction of sequence in nodes of this category in the context of the sequence occupied by the accession in the graph.

		Core			*Shell*		*Private*	
	Comp. Seq.	Percent nodes	Percent pan-genome	Percent path	Mb	Percent path	Mb	Percent path
AT1741	2.8	61.3	69.9	78.6	19.7	16.7	5.6	4.8
AT5784	3	61.7	71	77.3	19.8	16.5	7.5	6.2
AT6909	3	61.3	70.2	78.3	23.8	20.1	1.9	1.6
AT6911	2.6	60	70.1	78.3	16.1	13.6	9.6	8.1
AT7186	3.6	61.2	71.4	76.9	20	16.6	7.8	6.5
AT7213	3.1	61.4	70.2	78.3	20.5	17.3	5.2	4.4
TAIR10	2	60.9	69.5	79	23.5	20	1.1	0.9

The previously described pan-genome distribution was also described from a reference centric point of view. Here the *core* genome was, by definition, always traversed by the reference genome. From there onwards the amount of sequence that was also traversed by the *TAIR10* reference decreases with the number of accessions (Figure 3.8 (A)). The reference centric analysis of the graph showed that 60.9% of all nodes were traversed by

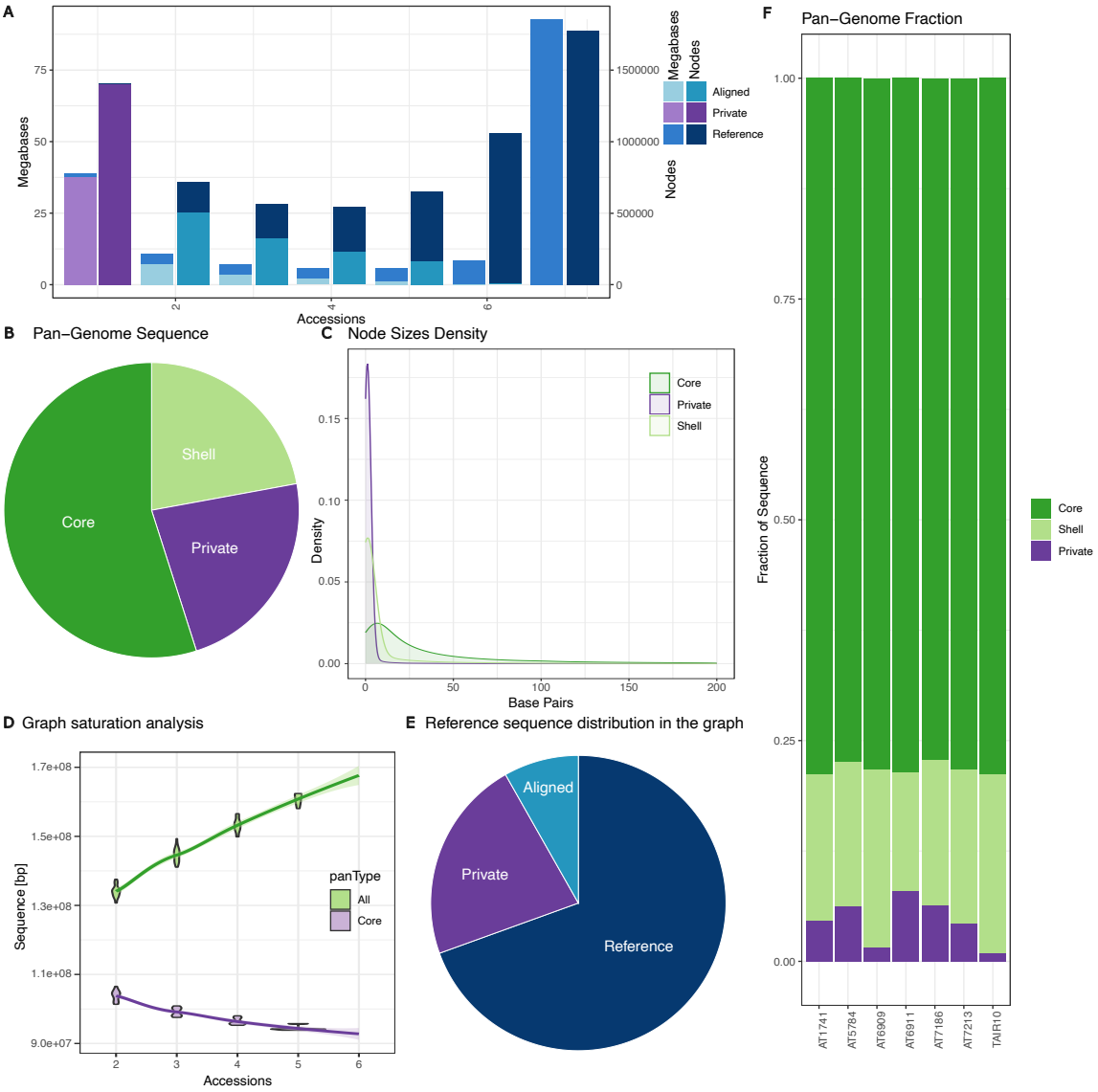

Figure 3.8: Graph Pan-Genome - Basic pan-genome statistics based on the genome graph. **(A)** Distribution of nodes and sequence (Mb) in the refernece based pan-genome. The colors are based on the presence of the reference genome in those nodes. *Reference*: The reference genome contains this sequence. *Aligned*: The sequence is present in at least two of the genomes, but not the reference genome. *Private*: The sequence is private to one of the *de-novo* assembled genomes. **(B)** Distribution of sequence in the pan-genome. *Core*: Sequence that is present in all genomes. *Shell*: Sequence that is present in at least two genomes, but not all. *Private*: Sequence that is only present in one genome. **(C)** Density plot showing the sizes of nodes that are part of the three pan-genome categories. **(D)** Saturation analysis based on the sequence in the graph. **(E)** Distribution of sequence in the pan-genome based on its occurrence in the reference genome. **(F)** Pan-genome classification of sequence in each of the assemblies and the reference genome.

the reference genome. 43.3% of them were *core* nodes in the pan-genome. The nodes that were not traversed by the reference genome could be divided into *private* (53.4%) and *shell* (46.7%) nodes. Despite the very similar fraction of nodes, the amount of sequence contained within these nodes differed drastically. 73.1% of the non-reference sequence remained *private* to one accession, while only 26.9% was part of the *shell* genome (Figure 3.8 (E)).

3.2.3 Graph SV calling

While small variation is easy to detect using conventional short-read based methods, complete assemblies enable us to detect large and more complex sequence variation. Here I focus on variable sequences larger than 50 bp. I first defined the sequence space that remained unaligned to the current *TAIR10* reference genome (Berardini *et al.*, 2015) and then intersected it with the variation called using *SyRI* (Goel *et al.*, 2019), as well as described its potential ancestry using a taxonomy analysis. I then used *panSV* (subsection 2.1.2) to detect variable regions from a pan-genomic point of view and describe them using the annotations I produced.

Non-reference sequence

Using the modified *panSV* approach, I detected 2,030,275 non-reference traversals. They summed up to a total of 74.7 Mb (mean size: 6.6 bp; median size 1 bp) of assembled sequence that had not been aligned to the reference genome. While almost 50% (838,655 traversals) of them were traversed by more than one genome, they only accounted for 3 Mb (mean: 2.7 bp; median: 1 bp) of the sequence space. Only 32,742 (1.6%) non-reference traversals were classified as *large variants* (≥50 bp). Nevertheless, they summed up to a total of 72 Mb (mean: 2.2 kb, median: 0.3 kb) of sequence, and thus accounted for the majority of affected sequence. 4,230 of these variants were traversed by multiple accessions. The number of traversing accessions decreased with increasing sequence length. The individual accessions contained an average of 627,605 non-reference traversals. *AT6909* had the lowest number of non-reference elements (16,411) and *AT6911* the

Table 3.11: Non-reference variants - Size and count of traversals in the graph that were not used by one of the *TAIR10* reference genome paths. The subset of variants that had a length≥50 bp were considered as large variants. Joined traversals are the number of unique traversals without a reference path in the graph.

	All Variants		Large Variants	
	Number	Mb	Number	Mb
AT1741	657,133	12.7	6,846	11.8
AT5784	769,435	15.6	7,799	14.5
AT6909	16,411	2.4	683	2.4
AT6911	895,625	16.5	9,386	15.2
AT7186	720,511	16.2	7,585	15.2
AT7213	706,517	15	7,475	14
Joined	2,030,275	74.7	32,742	71.7

highest (895,625). The average fraction of *large variants* was 1.6%. As *AT6909* had by far the highest fraction of large non-reference elements (4.2%), removing it reduced the mean fraction to 1%. The accession with the lowest amount of non-reference sequences was *AT5784* with 1%. (Table 3.11) Of these, on average 4.1% intersected with sequences that were classified as not-aligned by *SyRI*. A slightly higher fraction contained SVs over 50 bp detected by *SyRI* (5.2%). For general SVs the *AT6909* assembly contained the least amount (3.1%), while for non-aligned sequences it contained the most (4.4%) (Table B.5). The intersection of annotations with the non-reference regions revealed that on average 9.8% of the non-reference sequences contained genes. This value was the lowest in *AT6909* (7.8%) and the highest in *AT7186* (10.6%). The fraction of variants with TEs was higher. Here, on average, 21.9% of the non-reference regions contained at least one TE. Again, *AT6909* had the least TE containing non-reference sequences (11%). The most TE containing sequence was found in *AT6911*, with 24.4%. Almost all of the non-reference regions contained sequences that was annotated as repetitive (99.4%). Once again *AT6909* contained the least (96.9%), while the most was found in *AT1741* (99.9%) (Table 3.12).

Table 3.12: Non-reference annotation - Annotated features localized inside non-reference sequence stretches. The intersection was performed using *bedtools intersect* with an required overlap of at least 90% of the annotated feature. The percentage relates to the number of large non-reference variants of the accession.

	Genes		TEs		Repeats	
Accession	Count	Percent	Count	Percent	Count	Percent
AT1741	703	10.3	1,659	24.2	6,839	99.9
AT5784	797	10.2	1,895	24.3	7,789	98.7
AT6909	53	7.8	75	11	662	96.9
AT6911	926	9.9	2,286	24.4	9,369	99.8
AT7186	803	10.6	1,795	23.7	7,564	99.7
AT7213	772	10.3	1,799	24.1	7,464	99.9

In order to assess the source of non-reference sequence in my graph, I performed a taxonomy analysis on the large variants using *Kraken2* (Wood *et al.*, 2019). I was able to classify 66.8% of the large non-reference regions. The classified regions accounted for 97.5% of the sequence contained in the large non-reference regions (Figure 3.9 (A)). The majority of the non-reference traversals were originating from the *Arabidopsis* genus. 79.4% of them were assigned to *A. thaliana* itself, followed by *A. arenosa* (6.4%), and *A. lyrata* (5.5%). Other closely related genera, such as *Camelina* (1.4%), *Capsella* (0.9%), and *Brassica* (0.4%) were also represented (Figure 3.9 (B)).

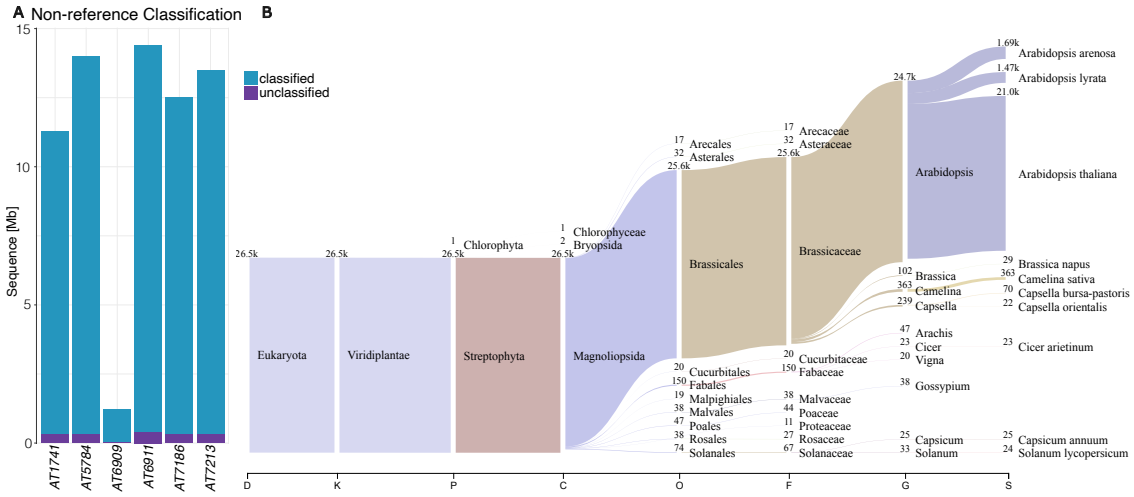

Figure 3.9: Non-reference sequences - *Kraken2* based taxonomy analysis of large ($\geq$ 50 bp) non-reference sequences from the graph based pan-genome. **(A)** Amount of sequence that was classified by *Kraken2* in each of the assemblies. **(B)** Sankey taxonomy representation of the most prevalent categories of the *Kraken2* results. The figure shows the number of sequences. For readability only the most common hits are shown in this representation.

Pan-genome based variant detection

A graph is a representation of the underlying alignment. Therefore it stores the variation of the aligned genomes and is able to describe their structure and nestedness better than reference based methods. Here I used the *panSV* (subsection 2.1.2), to access and describe the variable regions of the graph. In addition I intersected them with TEs, repeats, and genes that belong to non-standard orthogroups.
panSV detected a total of 2,561,981 variable regions from the graph. 1,792,643 of them belonged to the highest core level, which describes the number of genomes that are part of this region. The remaining regions detected by *panSV* were children of those bubbles and were nested within them (Figure B.4). 94.6% of all detected regions contained no nested children. The fraction of regions without children dropped to 91.8% when considering only regions with a core level of 6. The median size of regions, overall,

Table 3.13: *panSV* regions - Statistics of variable regions detected by *panSV*. Parent describes the percentage of regions that contain nested variants. Repeated describes the pervcentage of regions that are traversed multiple times by the same paths.

Level	Regions	Mean bp	% Parents	% Repeated
7	1792643	36	5.8	0.01
6	266389	84	8.2	0.01
5	173060	78	7.4	0.01
4	131502	82	5.9	0.02
3	113117	77	3.8	0.02
2	85271	89	0.01	0.04

and for the individual core levels was 1 bp. The mean size was strongly influenced by large regions. For core level 7 it was 36 bp. The other core levels averaged at 85 bp, ranging from 77 bp in core level 3 to 89 bp in core level 2. In rare cases, regions can have more traversals than its core level. This was the case for collapsed repeats and occurred in 0.01% of the variable regions. In regions with a low core level this behavior was a bit more common and increased to 0.04% in regions of core level 2 (Table 3.13). Using the length of traversals through a region I was able classify them as *SNPs*, *small variants*, or *large variants*, based on the criteria described previously. 76.9% of the regions were classified as *SNPs*, 13% as *small variants*, and 10.1% as *large variants*. These numbers differ from the results of *vg deconstruct* on the same graph (Table 3.5). Overall *vg deconstruct* detected fewer variants, and their fraction in the set differs. 11.6% of the variants detected by *vg deconstruct* were not overlapped with variants detected by *panSV*. The number of variable regions in each of the assemblies was very similar and ranged from 2,064,089 in *AT6911* to 2,113,235 in *AT5784*. On average 43.9 Mb of sequence was affected in each of the genomes. This amounts to over one third of the assembled genome sizes in each of the assemblies. The number of variable regions that did not belong to the highest core level was also comparable for each of the assemblies and ranged from 271,446 in *AT6911* to 320.592 in *AT5784* (Table 3.14).

Table 3.14: *panSV* paths - Number and size of variable regions per accession. The total number of regions, as well as the number of regions with a core level below 7 are shown.

Accession	# Regions	# Regions (CL <7)	Affected Mb
AT1741	2,110,064	317,421	42.3
AT5784	2,113,235	320,592	45.4
AT6909	2,103,188	310,545	44.1
AT6911	2,064,089	271,446	42.3
AT7186	2,102,357	309,714	45.7
AT7213	2,108,442	315,799	43.4

The intersection of TEs with the variable regions revealed that on average 0.2% of the regions contained at least one TE. Overall 51.8% of TEs per accession were located inside variable regions of the graph. When looking at genes only 0.1% of the variable region per accession contained at least one gene and on average just 6.1% of the genes annotated per accession were located inside these regions. On average 86.4% of the genes that were located in variable regions of the graph were also part of a non-standard orthogroup. Out of this set 24.8% of member genes per accession were located inside variable regions. In the total set of variable regions I identified 325 regions where the left and right anchor were the same node. These bubbles were labeled as repeat anchor regions. 240 of these regions overlapped with at least one TE (73.9%). In addition 133 of the repeat anchor regions contained a non-standard orthogroup gene (40.9%). 87.2% of those regions also contained at least one annotated TE. While only 0.01% of all variable regions

were repeat anchor regions, 1.7% of the regions that contained at least one TE had repeat anchors. For non-standard orthogroup gene-containing regions this fraction rose to 3.7%.

The reference gene *AT1G20400* was member of such a non-standard orthogroup. It encodes a hypothetical protein, that was annotated in the *araport11* reference annotation. This orthogroup contained seven genes that, in addition, were annotated in three of the *de-novo* assemblies and the *A. arenosa* outgroup annotation. No orthologs were found in *AT6911*, *AT7186*, and *AT7213*, but three copies were present in *AT5784*. All orthologs were located on chromosome one (Table 3.15). The variable region in the *TAIR10* reference genome (Berardini *et al.*, 2015) contained two annotated TE fragments alongside the annotated gene (Figure 3.10 (B), and at the insert site of one of the additional ortholog copies the left and right anchors were the same nodes, representing a repetitive anchor (Figure 3.10 (C)).

Table 3.15: Non-standard orthogroup - Copy numbers of members of orthogroup *OG0005825*.

Accession	Copies
TAIR10	1
AT1741	1
AT5784	3
AT6909	1
A. arenosa	1

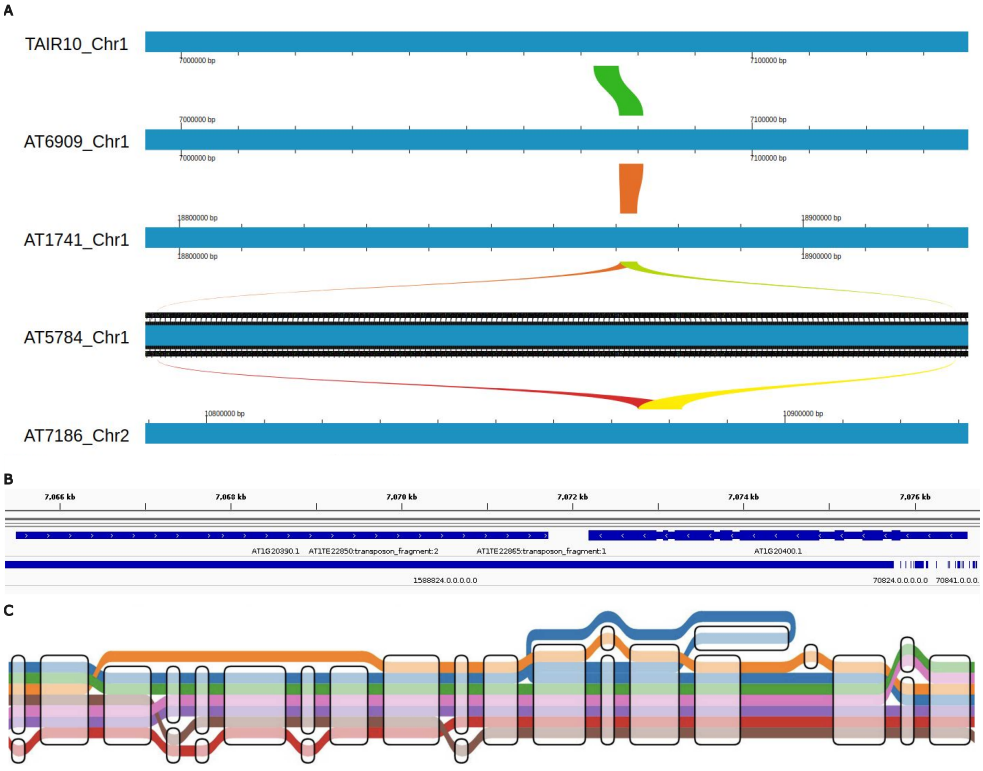

Figure 3.10: Non-standard orthogroups - Example of a non-standard orthogroup intersecting with a variable region in the graph. **(A)** Overview of the insertion sites of the orthologs. For simplicity only two of the sites in *AT5784* are shown. **(B)** Representation of the region in the *TAIR10* reference assembly. Showing the annotated features in *araport11*, and the variable regions with core level 7. **(C)** Layout of the graph at one of the insertion sites in *AT5784* (blue path).

3.2.4 Graph alignment evaluation

In order to access the full potential of a genome graph we need to be able to map reads to it. At the moment multiple mapping concepts and algorithms are available. I evaluated which one is the most suitable algorithm for the intended genotyping of nodes using short-read sequencing. To do so I compared four different mapping algorithms on a flat reference graph and a set of four graphs of increasing complexity. I mapped reads originating from the six assembled genomes as well as short-reads from six accessions randomly chosen from the 1001 Genomes Project (1001 Genomes Consortium, 2016) to the graphs (Table 3.16). I recorded the computational requirements and the mapping performance to compare the different graphs and mapping algorithms.

Table 3.16: Graph mapping test reads - Information on N90 read length, estimated coverage of the *TAIR10* reference genome, and the source of the reads used in the graph mapping evaluation.

Accession ID	N90	Estimated coverage	Source
AT1741	246	63.4	sixRef
AT1852	101	37	1001G
AT5784	248	69.2	sixRef
AT6680	101	18.2	1001G
AT6909	250	51.8	sixRef
AT6911	242	49.4	sixRef
AT7109	101	15.5	1001G
AT7186	235	151.7	sixRef
AT7213	248	73.5	sixRef
AT7384	101	17.4	1001G
AT7521	101	74.3	1001G
AT7568	101	81	1001G

The N90 read length of the six accessions with complete genomes were close to 250 bp, while the reads from the 1001 Genomes Project had an N90 length of 101 bp. The estimated coverage for the six accessions with complete genomes ranged from 49x to 152x, while for the 1001 Genomes reads it ranged from 15x to 81x. The creation of the target graphs is described in subsection 2.2.5. The *flat graph* contained the full length of the *TAIR10* reference genome (Berardini *et al.*, 2015) and no variation. It had a compression rate of 0 and a node degree of almost 1. The *VCF graph* had a compression ratio of 25.8% and a node degree of 1.5. The *chromosome graph* further compressed the sequence to 23.1% of the seven input genomes. The node degree decreased to 1.4. In the *linear graph* the compression decreased the input sequences to 22.9% of their original length, while the node degree remained stable at 1.4. The *complex graph* compressed the input sequences to 19.7% of their original size. The node degree dropped slightly to 1.37 (Table B.6).

The baseline for the comparisons was an alignment to the linear *TAIR10* reference genome using *bwa mem*. Here the memory consumption and run time scaled with the number of reads in each set. Using the conventional method, on average 96.2% of the reads were aligned to the reference genome, covering 95.4% (114.2 Mb) of the available sequence space. The first mapping algorithm, *vg map*, aligned on average 96% of the reads to the *flat graph*, covering an average of 95.4% (114.2 Mb) of the available sequence.

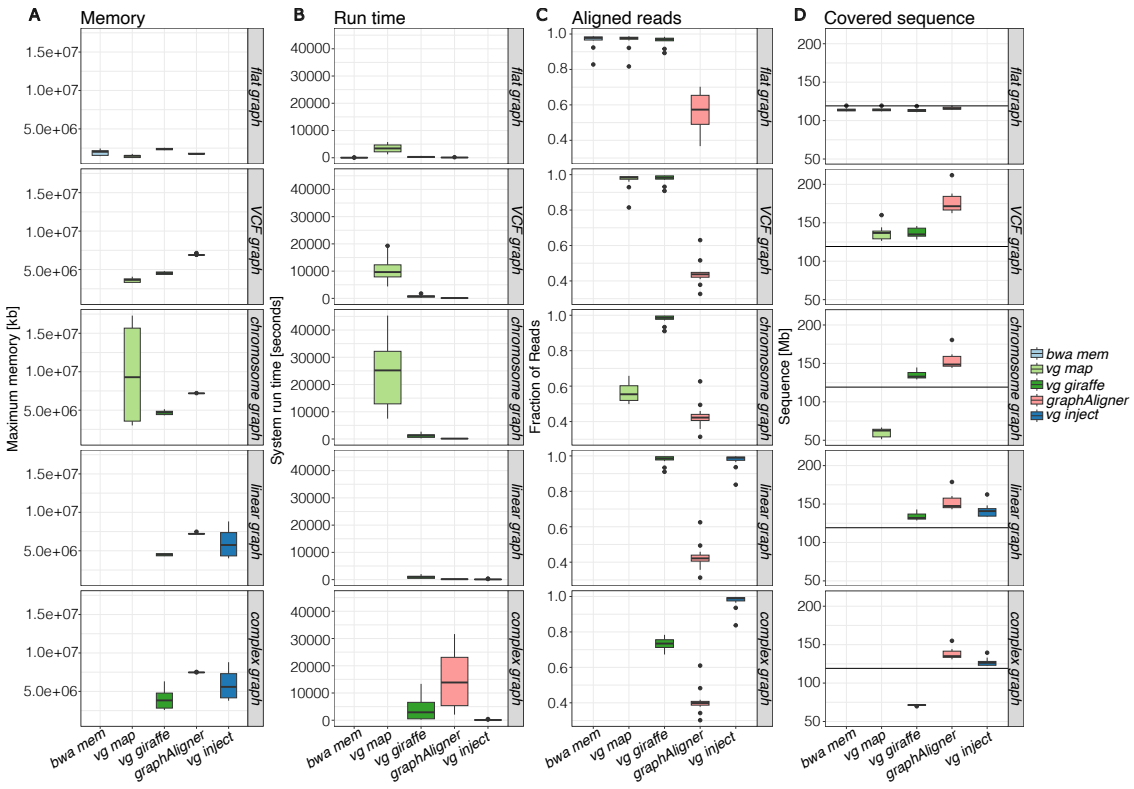

Figure 3.11: Graph mapping test - Statistics on the mapping performance of 12 short-read data sets to a flat reference and four graphs of increasing complexity. **(A)** Maximum resident memory requirement [kb] **(B)** System runtime of the alignments [seconds] **(C)** Fraction of reads aligned to the graph **(D)** Bases covered in the graph [Mb] - the line indicates the size of the *TAIR10* reference genome (119.2 Mb) (Berardini *et al.*, 2015).

The second algorithm, *vg giraffe*, aligned 96.1% of the available reads to 94.9% (113.6 Mb) of the genome. In contrast *graphAligner* was only able to place 56.8% of the reads, but covered 97.1% (116.9 Mb) of the available sequence space with it. On the *flat graph*, the mapping using *vg inject* is identical to the results of *bwa mem*. The *vcf graph* was mapped to by *vg map*, *vg giraffe*, and *graphAligner*. As no paths, except the reference paths are present in this graph, the performance of the *vg inject* based method would have been identical to the performance on the *flat graph*. The fraction of alignable reads increased by 0.5% for *vg map*, and 1.4% for *vg giraffe*. The fraction of aligned reads decreased by 12.4% for *graphAligner*. With the additional sequence added to the reference the amount of covered sequence in the graph increased for *vg map* and *vg giraffe*. *vg map* covered 136.6 Mb of the available sequence space, while *vg giraffe* mapped reads to 136.7 Mb. The amount of sequence covered by alignments from *graphAligner* was considerably higher (177.2 Mb). The first of the whole genome alignment derived graph (*Chrom graph*) saw a decrease of reads aligned by *vg map*. Only 56.2% of the available reads were mapped. The covered sequence decreased to 60.2 Mb. The amount of

reads aligned by *vg giraffe* increased by another 0.2%. The amount of covered sequence slightly decreases to 134.9 Mb. For *graphAligner* the fraction of aligned reads decreased further (43.1%), along with the covered sequence space (153.9 Mb). On the next graph (*linear graph*) *vg map* was unable to complete its run and failed. The *linear graph* was the first graph where the *vg inject* method could be used to project alignments into the graph. 97.1% of the reads were mapped by this approach and covered 72.4% of the graph sequence (141.5 Mb). The performance of *vg giraffe* did not change much with the slight increase in graph complexity. 97.7% of the reads were aligned, covering 134 Mb of sequence in the graph. The same was true for *graphAligner*. The fraction of mapped reads decreased by 0.04% and in total 152.6 Mb were covered. In the most complex graph the performance of *vg giraffe* dropped massively. Only 73.4% of the reads could be aligned by the algorithm, covering just 71.3 Mb of sequence in the graph. The performance of *graphAligner* kept decreasing. 41% of the available reads were aligned to cover 138.1 Mb of graph sequence. Meanwhile the fraction of reads injected into the graph using *bwa mem* and *vg inject* stayed almost identical at 97.1%. The amount of covered graph sequence decreased to 127.3 Mb (Figure 3.11) (Table B.7). A comparison of the computational resources required showed that on real graphs *graphAligner* always required the most memory. The consumption was independent of the graph complexity, the same was true for *vg giraffe*. Except for the *complex graph*, the memory consumption was almost identical over all graphs. *vg map* failed to map to the *linear*, and *complex graph* due to a massively inflated memory footprint. The memory consumption of the injection based method was among the higher ones in the comparison and constantly showed a wider range. In the runtime comparison, *vg inject* was among the fastest tools, while the other ones became slower with increasing complexity. Based on this analysis I decided to use the injection based method of *bwa mem* and *vg inject* to align reads to my genome graph (Figure 3.11).

3.2.5 Graph genotyping

The completeness of a graph-based pan-genome strongly depends on the selection of genomes used in the construction. By using it as a target for short-read alignments, the additional sequences in the graph extend the available sequence beyond that of the reference genome. The additional sequence can also be genotyped to expand the knowledge of variant frequencies. Furthermore the completeness of the graph structure can be estimated based on the amount of unmapped sequence, and variants can be called in an effort to increase their accuracy. In order to do so, I mapped a subset of the 1001 Genomes Project (1001 Genomes Consortium, 2016) short-reads to the six reference genomes graph and describe the expanded pan-genome of *A. thaliana* as well as the completeness of the constructed genome graph. I also describe the expanded pan-genome and called variants from alignments to the graph.

Read mapping

The quality filtering of the original 1001 Genomes Project short-reads (1001 Genomes Consortium, 2016) resulted in the removal of 295 accessions with reads that were too short or had insufficient coverage. The remaining 840 read sets belonged to ten different sub-populations, called admixture groups, which were identified based on the variants detected in the 1001 Genomes Project (Table B.8). The short-reads in this subset were produced by four different laboratories (Figure 3.12 (A)). They were mapped to a combined fasta sequence of the six *de-novo* assemblies and the *TAIR10* reference genome (Berardini *et al.*, 2015) with varying completeness. The median percentage of mapped reads was 98.8% (mean: 96.8%). 696 accessions had a higher percentage of mapped reads, than the mean. 350 accessions had more than 99% of their reads mapped. The lower end of the distribution was much broader. The lowest accession only mapped 43.1% of its reads. Only 17.1% of the sets mapped less than the mean fraction of mapped reads. These accessions contained 34.7% of all read sets sequenced at the Salk Institute (Figure 3.12 (B)). On average 1.2% more reads were mapped to the graph compared to the *TAIR10* reference genome.

The reads that remained unmapped to the graph were subjected to a *Kraken2* taxonomy analysis. Here between 6% and 99.2% (mean: 80.8%; median: 88.4%) of the reads remained unclassified. The remaining reads were identified as *viridiplantae* in

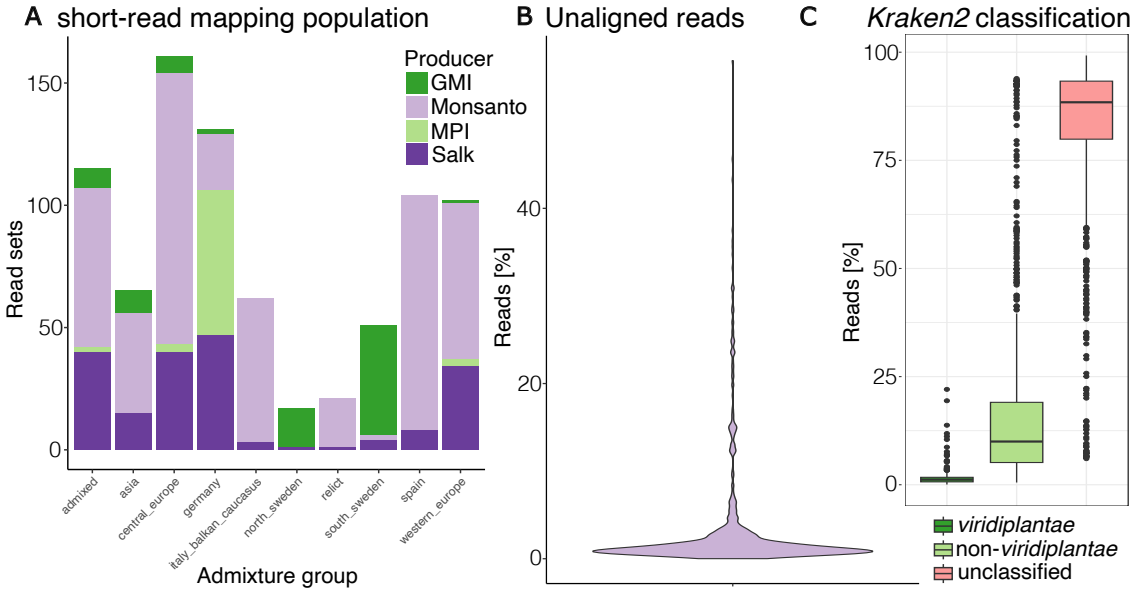

Figure 3.12: Short-read mapping population - **(A)** Distribution of admixture groups in the subsetted 1001 Genomes Project mapping population. The lines have been coloured by the laboratory that produced the individual short-read set **(B)** Fraction of reads, in each short-read mapping accession, that did not align to the genome graph. **(C)** Percentage of classification of unaligned short-reads by *Kraken2*. Split into reads remained unclassified, reads classified as *viridiplantea*, and reads classified as any other clade.

on average 1.4% of the cases (median: 1.2%), or as *non-viridiplantae* in 17.9% of the cases (median: 10%). The fraction of reads assigned to *viridiplantae* ranged from 0.05% to 22.05% (Figure 3.12 (C)). This distribution shifted in the set of accessions that aligned a lower percentage of reads than the mean. In these accessions fewer reads remained unclassified (mean: 71.4%; median: 85%), and fewer reads were calssified as *viridiplantae* (mean: 0.4%; median: 0.3%). In contrast, more reads were classified as *non-viridiplantae* (mean: 28.1%; median: 14%).

Genome size estimation

Table 3.17: SixRef short-read mappings to the Graph - Statistics on the sequence covered by short-reads from the three accessions that are part of the genome graph, and the mapping population. The additional fraction is in comparison to their representation in the graph.

Accession	Graph [Mb]	Covered [Mb]	Additional [%]
AT5784	119.9	122.8	2.4
AT6909	118.4	120.3	1.6
AT6911	118.4	129.8	8.8

Using the node coverage information I was able to estimate the collapsed genome size of each short-read accession. The least graph sequence was covered by the Western european accession *AT9862* (106.4 Mb), while the most sequence was covered by *AT6933* (136.7 Mb), a Spanish accession. The mean covered graph sequence was 121.2 Mb (median: 121.1 Mb). In total 614 accessions covered more sequence in the graph, than the length of the current *TAIR10* reference genome (119.2 Mb) (Berardini *et al.*, 2015) and 706 more than the collapsed reference sequence as it is represented in the graph (117.3 Mb) (Figure 3.13 (A)). Three out of six accessions used to build the graph were also represented in the set of short-reads mapped to the graph. All three short-read sets aligned to additional sequence beyond the nodes that represented this accession in the graph. The lowest amount of additional sequence was found in *AT6909*. Here, just 1.6% of additional sequence were covered by alignments to the graph. *AT5784* aligned to 2.4% of additional sequence and *AT6911* to 8.8% (Table 3.17).
In addition to considering the sequence covered by reads mapped to the graph, I also estimated the sequence that had not been represented in the graph using the unmapped reads. The median amount of estimated additional sequence is 324 kb (mean: 318 kb). The set of genomes that showed an abnormal amount of unmapped reads also had the highest amount of estimated additional sequence (mean: 1.42 Mb; median: 1.13 Mb). For the majority of accessions the amount of estimated additional sequence did not correlate with the fraction of unmapped reads.

I categorized the mapped accessions by admixture group. Admixture groups are subpopulations of *A. thaliana* that were defined based on their genetic similarity in the 1001 Genomes Project (1001 Genomes Consortium, 2016). The individual groups showed different median estimated genome sizes. Both Swedish groups covered the most graph

sequence. The median of accessions classified as Southern Swedish was 125.4 Mb, and the median of Northern Swedish accessions was 125.2 Mb. The lowest median amount of covered sequence was found in the Asian admixture group (120 Mb). Together with the

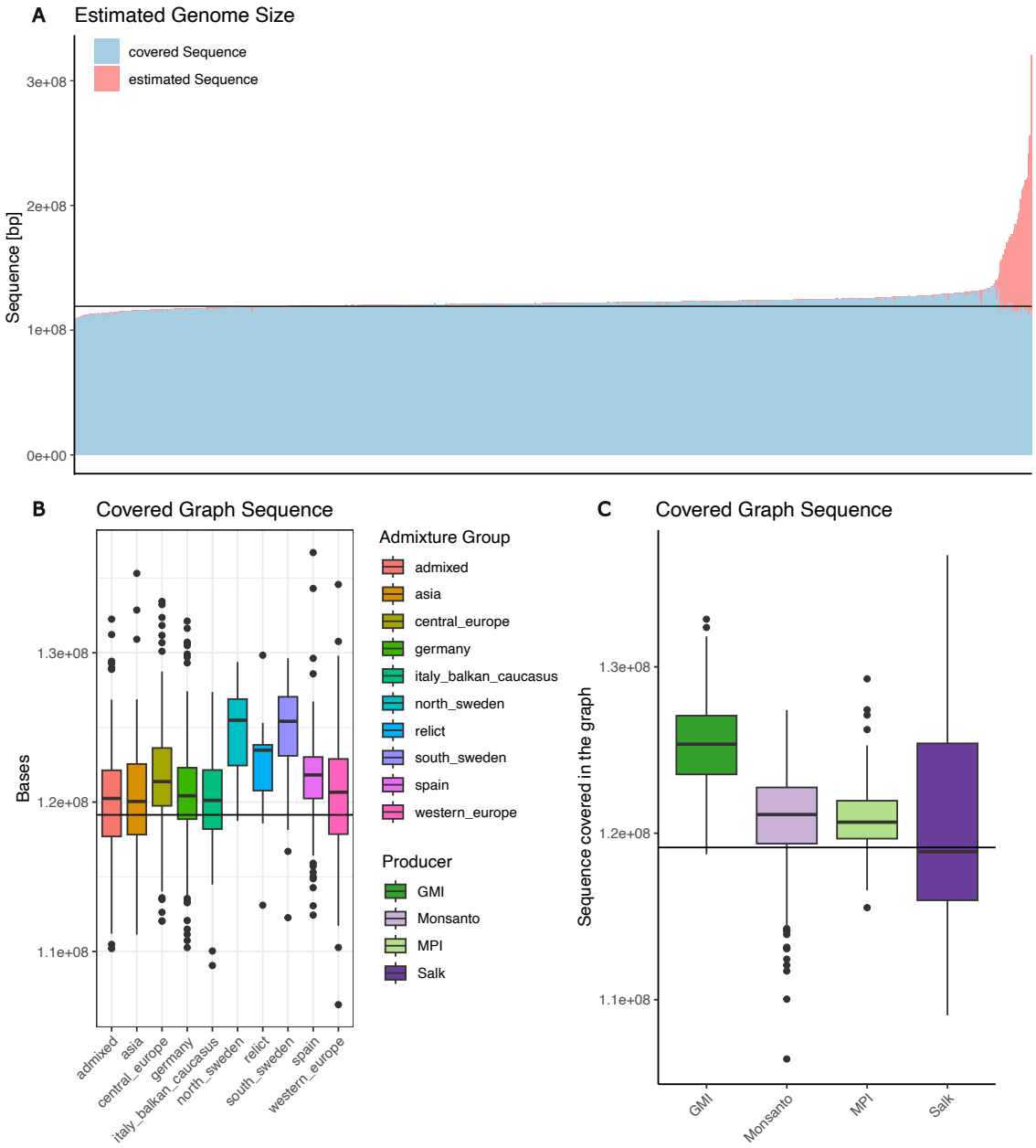

Figure 3.13: Genome size estimation - Estimated genome sizes of the mapping population. The calculation is based on the amount of covered sequence in the graph, and a k-mer based sequence size estimation of the unmapped reads. **(A)** Estimated size of every accession. The black horizontal line indicates the size of the *TAIR10* reference genome. Estimated genome size of the accessions, grouped by their **(B)** admixture groups, and **(C)** the lab where they had been sequenced.

German admixture group, of which *AT6909* is a member, the Asian admixture group had an estimated genome size closest to that of the *TAIR10* reference genome (Figure 3.13 (B)) (Table B.9). A one-way ANOVA correlation analysis showed that the amount of covered sequence in the graph correlated with the assigned admixture group (p=1.77e-13). A even higher correlation could be observed between the covered sequence in the graph and the lab that sequenced the accession (p=1.16e-16) (Figure 3.13 (C)).

Alignment based pan-genome

By combining the coverage information obtained from all accessions, I could explore the pan-genome of the sequenced population. The graph sequence based pan-genome showed a U-shaped distribution (Figure 3.14 (A)). Based on the threshold of other pan-genome studies, I classified sequence as *private* if it attracted mappings from fewer than 10% of all accessions (< 84). Using this definition, 10% of the graph sequence was classified as *private*. 29.8% were considered as *shell* genome, being mapped to by 85 to 755 individual genomes. 55.2% of the sequence stored in the graph were classified as *core*. 5% of the sequence in the graph was not covered by any reads (Figure 3.14 (B)).

Table 3.18: Mapping based pan-genome - Pan-genome distribution of the accessions mapped to the graph. The *core* category contains nodes that are mapped to by at least 90% of the sequences. Nodes that are mapped to by 10% or fewer accessions are classified as *private*.

	# Nodes	Mb	Mean bp	Median bp
Core	2,255,312	93.3	41.35	9
Shell	3,488,995	50.3	14.41	1
Private	933,257	16.8	18.05	1
unmapped	17,242	8.4	486.68	79

For *private* and *shell* nodes the median and mean node sizes were very similar. The mean node size for *private* nodes was 18 bp (median 1 bp), the mean size of *shell* nodes was 14 bp (median 1 bp). For *core* nodes the mean size increased to 41 bp (median: 9 bp). Nodes that remained unmapped had the largest mean (487 bp) and median (79 bp) (Table 3.18). The sequence in the set of nodes that remained uncovered consisted almost entirely of Ns (97.62%).

In addition to the description of the mapping based pan-genome I explored the predictive power of the graph itself. I compared the pan-genome categories of nodes in the graph with their category in the mapping population. 48.4% of the nodes that were categorized as *core* in the graph were also categorized as *core* in the mapped population. The remaining nodes were attributed to *shell* (40.7%), *private* (10.8%), and uncovered nodes (0.1%). Nodes that had been categorized as *shell* in the graph were also categorized as *shell* by the mapped population in 67.8% of the cases. 22.5% were categorized as *core* based on the mapping population, despite belonging to the *shell* genome in the aligned population. 10.8% were assigned as *private* in the graph, and 0.09% as uncovered. For the *private* nodes of the graph the picture changed and the *private* mapping population did not form the largest intersecting group (27.8%). Most of the *private* nodes in the graph

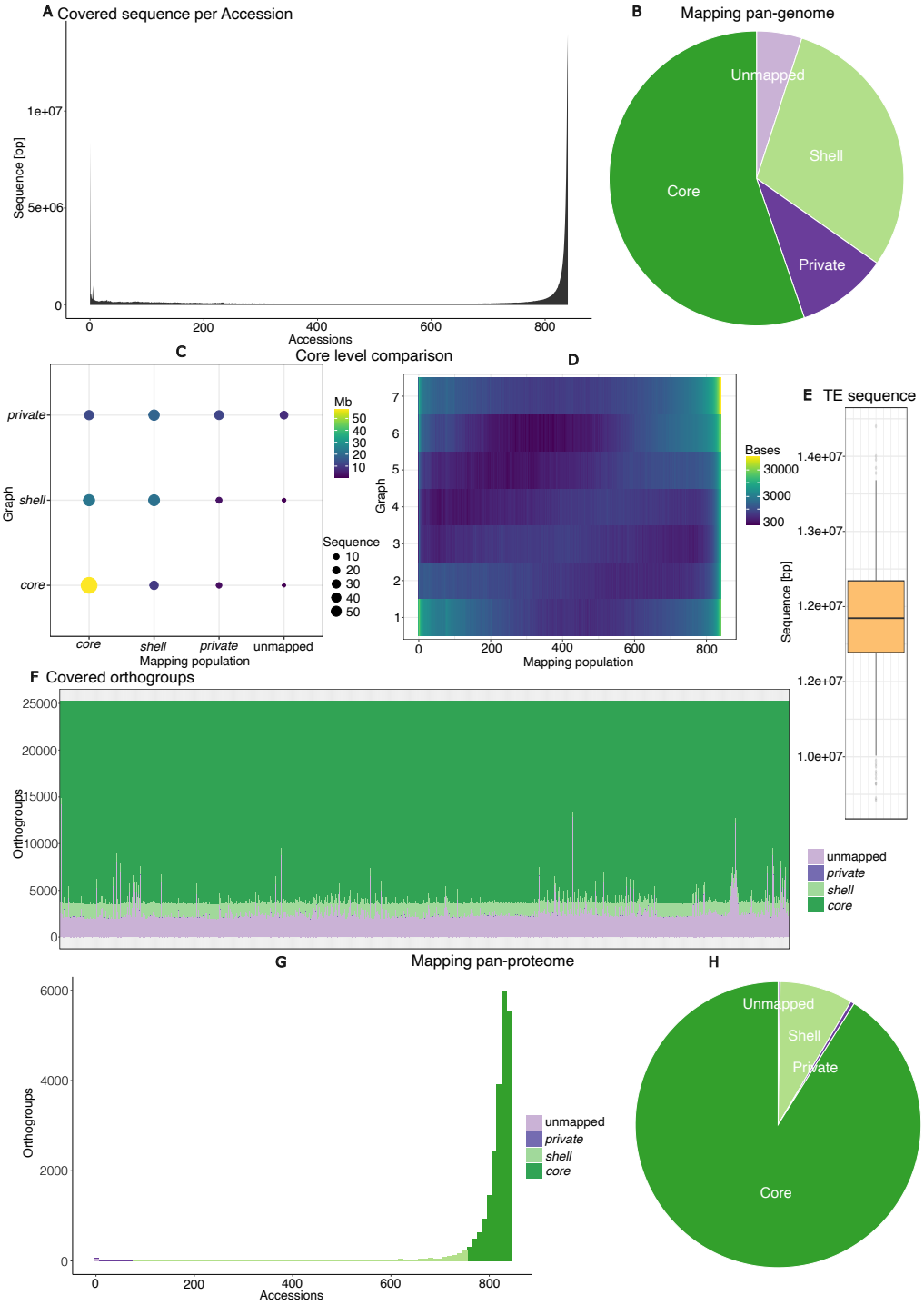

Figure 3.14: Pan-genome & pan-proteome analysis - Graph based pan-genome analysis of the mapping population. **(A)** Distribution of sequence frequencies in the pan-genome of the mapping population. **(B)** Distribution if pan-genome sequence. **(C)** Comparison of the pan-genome levels between graph and mapping populations. **(D)** Heat map of graph node traversals and mapping frequency. **(E)** Amount of TE space in the graph covered by each of the mapped accessions. **(F)** Number of orthogroups covered by each of the accessions in the mapping population and their attribution to the pan-proteome levels. **(G)** Distribution of orthogroup frequencies in the mapping based pan-proteome. **(H)** Pie plot of the fraction of orthogroups assigned to each pan-proteome level.

were attributed to the *shell* genome of the mapping population (47.4%). The *core* mapping population intersected with 23.9% of the *private* graph nodes, and 0.9% remained uncovered. When considering the number of bases, instead of the number of nodes the picture changes. 82.7% of the *core* sequence in the graph were also identified as *core* in the mapping population. In the *shell* genome only 45.4% were also part of the *shell* genome of the mapping population, while 47.9% were part of the *core* genome. 14.9% of the *private* graph pan-genome bases remained uncovered in the mapping population. 36.5% of the bases were part of the *shell* pan-genome of the mapping population, 25.5% of the *core*, and only 23.1% were *private* in both sets (Figure 3.14 (C)). The comparison of the pan-genome level of each node in the graph and the mapping population showed a diagonal line that indicates a correlation between the two (Figure 3.14 (D)) (Table B.10).

Alignment based pan-proteome

In addition to the raw node space I was also able to describe the distribution of the features annotated in the genomes that were used for the graph construction. Here I describe the distribution of TEs and genes in the mapping population. As an orthogroup-like assignment of TEs is not trivial, I resorted to projecting the TEs onto the graph and simply annotated regions of the graph as containing a TE in at least one accession. In total, I annotated 19.2 Mb of the graph as TEs. This was 2.3 kb more than the average amount of TEs in the *de-novo* accessions. None of the accessions from the mapping population came close to covering all TEs in the graph. The accession with the most covered TE sequence was *AT6933* (14.4 Mb). The accession that covered the least TE sequence was *AT7068* (9.4 Mb). The mean, and median, TE sequence covered by accessions mapped to the graph was 11.8 Mb (Figure 3.14 (E)). Only 55,710 nodes that contain TEs were covered by reads from more than 90% of the mapped accessions. This sums up to a total of 657 kb of TE sequences. The absolute majority of these nodes (53,728) were covered at more than 1.5 x median coverage. In contrast, 209,227 nodes (4.9 Mb) from the set of TE nodes were covered by less than 10% of the mapped accessions. When only considering nodes that were annotated as TEs and larger than 500 bp, a clear pattern of covered and uncovered nodes could be observed that is conserved for most of the accessions. Notable in this distribution were nodes that had a consistently higher coverage, up to 50 fold of the median coverage of the accession (Figure B.5).
In contrast to TEs, I was able to utilize the orthogroup assignment in the graph to collapse nodes that contained genes that were part of the same orthogroup. Based on the coverage of genes and respectively their orthogroup I was able to describe their distribution in the wider population. In total 25,271 orthogroups were constructed based on the assembled, and annotated genomes and at least one member of an orthogroup in the graph had to be covered to at least 80% of its sequence length in order for this orthogroup to be considered as present in the mapped accession. 1,557 orthogroups were present in every accession, based on this analysis. Using the 10% *core* threshold, established for the node based analysis, the *core* category contained an additional 8,827 to 20,215 orthogroups

per accession (mean: 19,729 OGs; median: 20,055 OGs). *Private* orthogroups were rare. No orthogroup was present in only a single accession. Overall a mean, and median of 4 orthogroups per accession were considered as *private* in the mapping population The numbers ranged from 0 orthogroups, in 90 of the mapped accessions, to a maximum of 37 private orthogroups in *AT6911*, one of the genomes that was part of the graph. The other two accessions from the sixRef set, that were present in the mapping population contained 25 *private* orthogroups (*AT5784*), and 19 *private* orthogroups (*AT6909*), placing them in the top 10 mapped accessions with the most *private* genes. The number of *shell* orthogroups was also comparatively small. Here the number of orthogroups ranges from 583 to 1,661 orthogroups, with a mean of 1,355 orthogroups (median: 1,381) (Figure 3.14 (F)). The z-Score analysis of the orthogroup copy number did not reveal a clear pattern (Figure B.6).
The pan-proteome occurrence of the 23,909 orthogroups, detected in the sixRef genomes, in the mapping population exhibited a similar distribution as the node occurrence. The absolute majority of the orthogroups belonged to the *core* pan-proteome (91.1%). Only 8.3% were assigned as *shell* orthogroups and a total of just 102 orthogroups (0.4%) were *private* in the pan-proteome, that means covered by alignments of less than 10% of the accessions in the mapping population. 56 orthogroups (0.2%) were not covered by any of the accessions of the mapping population (Figure 3.14) (G) & (H)). 53 of them were *private* to one of the sixRef genomes, two orthogroups were shared among two sixRef genomes and a single orthogroups was present in five out of six of the sixRef annotations. In the pan-proteome analysis of the sixRef *de-novo* annotations I identified 148 *core* orthogroups that had no contributing transcript from the *araport11* reference annotation, but a transcript from the *A. arenosa* outgroup annotation. 136 of them were also *core* orthogroups in the mapping population and the remaining 12 were *shell* orthogroups. The frequency of the *shell* orthogroups ranged from 371 to 750 accessions (mean: 605.6 ; median: 630.5).

Variant calls

With the variant-enriched genome graph as a new reference structure, I could call variants and investigate the impact of the additional sequences on their quality. I called variants from each of the 840 accessions in the mapping population. They were each intersected and compared with the original variants called from the 1001 Genome Project (1001 Genomes Consortium, 2016). In total, I called 1,844,756 variable positions that contain 3,222,637 different alleles. Each position is variable in, on average, 241.8 accessions of the mapping population (median: 161), and has 1.8 different alleles (median: 1). The majority of the alleles were categorized as *SNPs* (43.2%), in addition 16.6% of *small variants*, and 40.2% of *large variants* were called. The median variant size was 1 bp (mean: 79.5 bp) (Figure 3.15) (A)).The *SNPs* are distributed along the whole chromosome, while the *small*, and *large variants* are clustering around the pericentromeric region (Figure 3.15 (D)). On average 134.9 accessions contributed to an

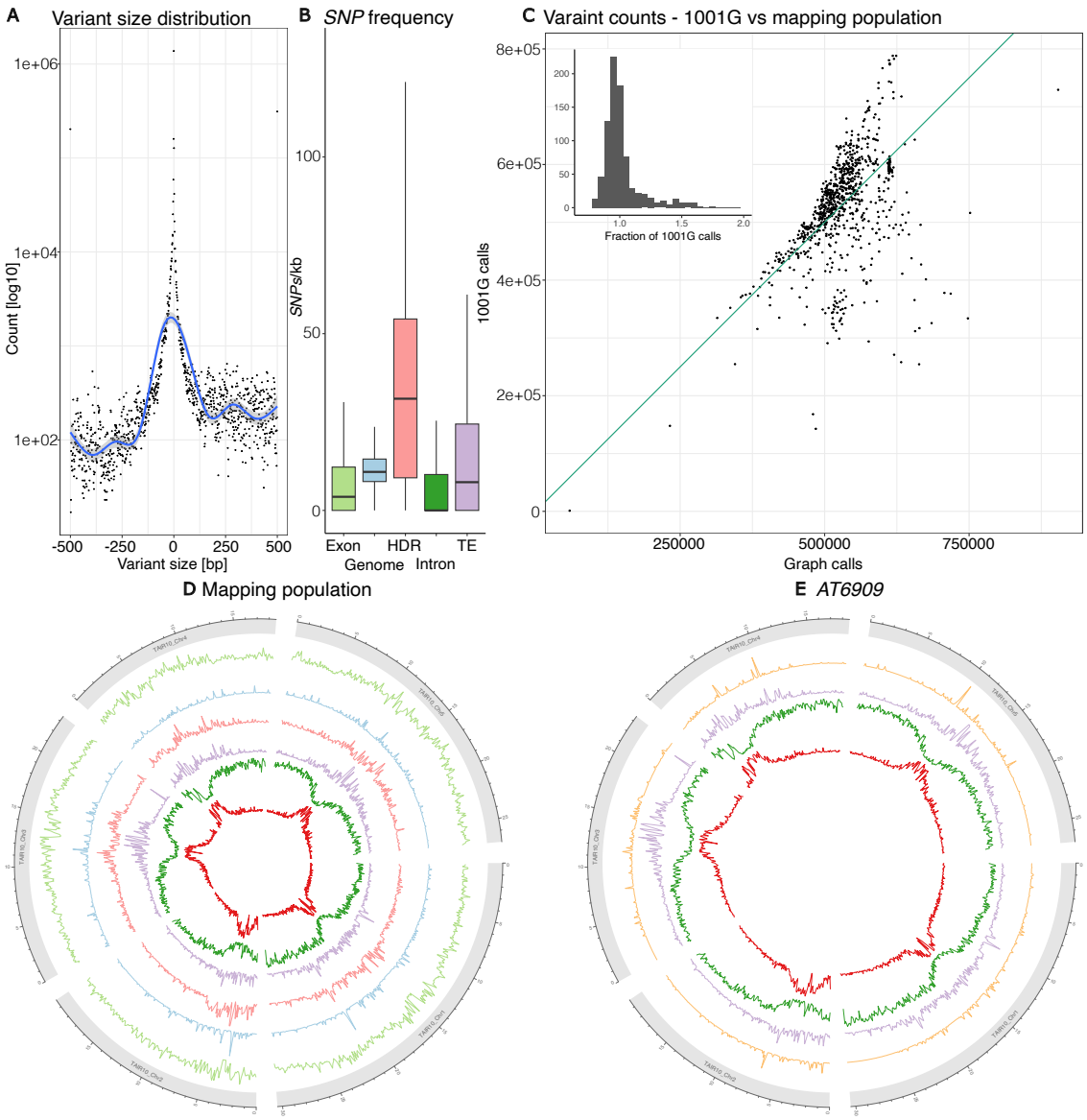

Figure 3.15: Short-read variant calls from the graph - **(A)** Distribution of sizes in the combined variants that were called from the short-read mappings to the graph. **(B)** Frequency of called *SNPs* per kb in different regions of the *TAIR10* reference annotation. **(C)** Comparison of the number of variants called from the graph with the number of variants in the 1001 Genomes Project data for each accession. In addition the relation between the two sets for each accession as the fraction of the number of calls in the 1001 Genomes Project is shown. **(D)** Circos plot of the relation between variants and the genomic features for the complete mapping population. The tracks, from the outside in show: frequency of *SNP* calls (light green), frequency of *small variants* (light blue), frequency of *large variants* (light red), density of highly diverged regions (HDR) (purple), gene density (green), TE density (red). **(E)** Circos plot of the density of heterozygous variants called from the reference accession (*AT6909*), in comparison with features of the *TAIR10* reference genome. The tracks, from the outside in show: frequency of heterozygous variants (orange), density of highly diverged regions (HDR) (purple), gene density (green), TE density (red).

allele (median: 16). The number of accessions greatly differed between the variant size categories. While *SNPs* (mean: 209.5; median: 121) and *small variants* (mean: 208.1; median: 131) contained a similar number of accessions per site, *large variants* were shared between very few accessions (mean: 7; median: 1). On average 2.3 accessions per *large variant* were heterozygous (median: 1). In *SNPs* the heterozygosity was at 32.1 accessions per variant (median: 12), and at 62.2 in *small variants* (median: 26). In total 81.3% of the sites were heterozygous for at least one accession from the mapping population. Again *SNPs* (91.7%) and *small variants* (97.6%) had a very similar fraction of heterozygous variants. *Large variants* were heterozygous in at least one accession for 62.5% of the sites (Figure 3.16 (G)).
Variants were not distributed uniformly throughout the sequence. The full sequence of the *TAIR10* reference genome (Berardini *et al.*, 2015) contained an average of 11.5 *SNPs* per kb. This number increases to 15.2 *SNPs* per kb in the regions that were identified as highly diverged (HDR) by *SyRI* in the analysis of the six *de-novo* assembled genomes. The variant frequency also differed between features of the *araport11* reference annotation (Cheng *et al.*, 2017). TEs had an increased number of 17.5 *SNPs* per kb, while it was decreased for genes (10.2 SNPs/kb), the difference between introns (8.2 SNPs/kb) and exons (9.6 SNPs/kb) being minimal (Figure 3.15 (B)).
An analysis based on the *A. thaliana* pseudo-heterozygosity study (Jaegle *et al.*, 2021) yielded 672,654 *SNPs* that were called inside genes of the *araport11* reference annotation. Of them 11,677 were heterozygous and called in more than 5% of the mapping population. Each of the positions was present in on average 814.8 samples (median: 814). In the filtered *SNPs*, on average 4.6% of the lines were heterozygous at a site.
The individual accessions contained between 58,805 (*AT6909*) and 994,236 (*AT6911*) variable positions (mean: 530,959; median: 529,204). On average 75.6% of the called variants were *SNPs* (median: 75.5%) followed by 22.6% of *small variants* (median: 22.7%), and 1.8% of *large variants* (median: 1.8%) (Figure 3.16 (F)). The individual number of called variants is comparable to the number detected in the original 1001 Genomes Project call set (mean: 1.1x 1001G calls; median: 0.98x 1001G calls) (Figure 3.15 (C)). The accession *AT9887* contained the 0.78x of the 1001 Genomes project calls, while the reference accession *AT6909* contained 54.2 times the amount of variants compared to the 1001 Genomes Project call set. The next highest comparison was in AT7461 with 3.4x. On average 70.1% of the variants from the original 1001 Genomes Project calls were re-called from the graph. 58.3% of them were exact matches (median: 57.1%). The remaining 11.8% were overlapping variants that were no exact matches (median: 11.3%). 29.9% of the variants from the original 1001 Genomes Project could not be re-called from the graph (median: 31.5%). The accession with the lowest recall rate was the relict accession *AT9905*. 46.9% of the called variants could not be recovered. In the accession *AT5784*, which is also part of the graph, only 9.2% of the variants could not be re-called (Figure 3.16 (I)). I checked the coverage of the positions in the graph that could not be recalled from the graph. The reference sequence was covered by the alignments in on average 90.9% of the cases without a variant call (median: 91%).

While the majority of the combined sites were heterozygous for at least one accession, homozygous variant calls were the norm in each of the accessions (mean: 84.8%; median: 86.9%), I nevertheless called different types of heterozygous variable positions. On average 14.2% (median: 12.1%) of the calls were heterozygous, and contained a reference allele. The remaining 1% of heterozygous variation contained two non-reference alleles (Figure 3.16 (H)).
The reference accession *AT6909* was the outlier in all of the analysis. It contained, by far and as expected, the fewest variants (58,805), and the distribution of variant sizes also differed. It contained the least *SNPs* (59.6%), and the most *small variants* (38.1%), and *large variants* (2.3%). 68.4% of the variants from the 1001 Genomes Project were exact re-calls in the graph, 10.9% were overlapping, and 20.7% could not be re-called. Only 5.6% of the variants were homozygous. 1.5% of the heterozygous variants contained two novel alleles. The heterozygous variants were enriched in the HDR. The complete genome contained 0.5 heterozygous variants per kb, while the HDR contained 6.1 heterozygous variants per kb (Figure 3.15 (E)).

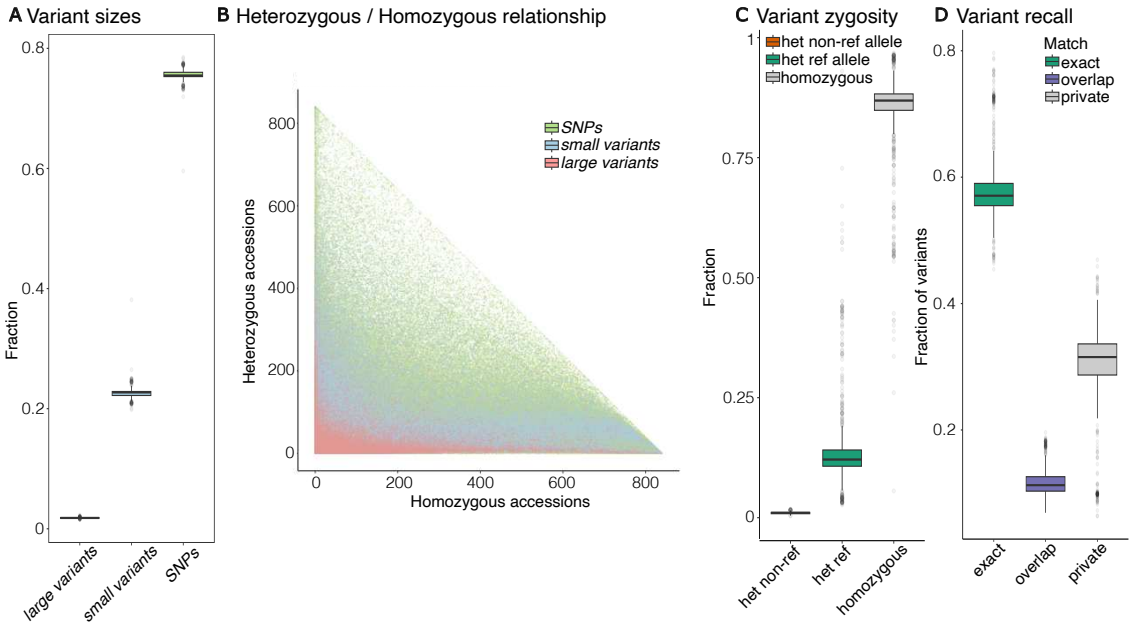

Figure 3.16: Short-read variant statistics - **(A)** Comparison of the fraction of variants attributed to the three size categories in each of the accessions from the mapping population. **(B)** Relationship between the number of homozygous and heterozygous variant calls for each accession, coloured by the different size categories. **(C)** Distribution of heterozygous variants in each of the accessions of the mopping population. **(D)** Fraction of variants on each accession that either intersect, overlap, or are private in the comparison of the 1001 Genomes Project call with the variant calls from short-reads aligned to the genome graph, in each accession of the mapping population.

Chapter 4

Discussion

The aim of my dissertation was twofold: (1) to explore the power of additional full-length genomes for a richer understanding of genome variation in this species, and (2) to learn how additional genomes and genome graphs, as a novel reference structure, can be leveraged to provide more value to existing short-read data sets, such as the one from the 1001 Genomes Project (1001 Genomes Consortium, 2016). This required additional genome assemblies to enlarge the available sequence space as well as new methods to build genome graphs and use them. As we have discussed before, current reference based analysis is strongly influenced by the available reference genome and its quality. Missing, or misrepresented sequence causes a reference bias. This bias can be reduced by adding additional alleles to the reference genome. In an effort to add missing sequence space six *de-novo* assemblies of *A. thaliana* were created. The comparison of these highly contiguous assemblies underlined not only the high synteny of the *A. thaliana* population, but also the power of whole genome assemblies for a more unbiased structural variation detection. A comparison of variants detected in the reference framework, using different methods showed the high variability of those calls. I deployed a new, combinatorial approach to annotate the assemblies. I created a pan-proteome based on this annotation, that showed a similar synteny. I was able to describe the changes in orthogroup sizes and discover a set of potentially ancestral genes that are missing from the current reference annotation. Using the *de-novo* assemblies I then constructed a genome graph to represent the previously unknown and unrepresented sequences. I was able to show that we can already represent a large portion of the core genome of *A. thaliana*, while we are still lacking representation of the shell genome. This was done by aligning a set of short-reads from 840 accessions to the complex, whole-genome derived genome graph, using a method I established. I highlighted the existing challenges of constructing a graph from multiple whole genome alignments by describing the portion of the graph not aligned to the reference. I used *panSV*, a novel graph based, reference free, variant detection tool, developed for this thesis, to discover traces of the mobilome in the graph itself. In addition to enlarging the pan-genome and pan-proteome I was able to show that using a genome graph as an alignment target for reads can drastically reduce the reference induced pseudo-heterozygosity seen in short-read analyses (Jaegle *et al.*, 2021).

4.1 The sixRef pan-genome

The *de-novo* assemblies of the six accessions are highly contiguous, to a degree where multiple chromosome arms are assembled as one contig. The chosen approach of long read sequencing, short-read polishing, with extra optical maps to ensure correct scaffolding gave me the opportunity to detect and place structural variants with a higher reliability. Despite the fact that, based on the scaffolded sequence length, we only exceeded the reference genome size in three out of six assemblies, we can still conclude, that we were able to assemble highly contiguous genomes du to the fact, that we were able to scaffold full chromosome arms without major stretches of Ns. The centromeric regions remain the problematic areas of the assemblies. Using a pairwise comparison with the reference I was not only able to show the high conservation of the *A. thaliana* genome structure, where 77.2% of the reference genome is syntenic with all six assembled genomes, but also observe previously described structural variants in the assemblies (Zapata *et al.*, 2016; Rowan *et al.*, 2019; Jiao and Schneeberger, 2020). PAVs detected from the alignments are unbiased in size. While in short-read based analysis the results were always biased towards the detection of deletions over insertions (Kosugi *et al.*, 2019; Ho *et al.*, 2020), I can now observe 0.9 Mb more inserted sequence over deleted variation. This is a direct result of the comparison of whole genome assemblies with the reference genome, as those loci missing from the reference are now assembled in the reference. In addition, the increased resolution of repeat structures easily adds to the additional sequence space that can be detected as insertions. While the current surge in telomere to telomere assemblies allows an even better representation of structural variants in genomes (Gonzalez de la Rosa *et al.*, 2021; Giguere *et al.*, 2021; Wang *et al.*, 2021), the slightly incomplete six *de-novo* assemblies already increase the power to detect large SVs and represent them in their sequence context.
One of the accessions chosen for assembly was *AT6909*. The current *TAIR10* reference genome (Berardini *et al.*, 2015) is based on the same accession and thus our assembly has a considerably lower number of variants detected. Interestingly the distribution of detected variant sizes shifts from a majority of *SNPs*, in all the other accessions, to larger variants. This is the direct result of the changes that stem from the advances in genome sequencing and assembly methods, as already described by (Wang *et al.*, 2021). In addition to the shift in variant sizes I can also observe a shift in SV type, as copy number changes are far more common in this comparison, than in all the other accessions. As the assembly is highly similar to the reference, real structural rearrangements are rare and the observed copy number changes reveal the sequence of the previously hard-to-assemble parts taht were missing in the reference genome.
I had initially planned to intersect three different variant sets of the six accessions in an effort to create a reliable set of variants for method validation. This effort failed as the variations that were no SNPs were far too dissimilar. Nevertheless a valuable lesson on the method-inherent biases was learned, as variants called by different methods were hard to compare. Different approaches rely on fundamentally different concepts to de-

tect variants. While short-read based methods are sufficient to describe small base pair differences, changes in coverage distribution, and insert sizes, they suffer from the previously described reference bias. These methods are mostly blind to the syntenic concept that can be employed by whole genome comparisons that are able to describe large-scale variation. The assembly based methods are not entirely without their issues either, as they suffer, for example, from the exclusion of contigs in the analysis, as in the *SyRI* analysis, or struggle with the detection of inter-chromosomal rearrangements. In addition the classification of certain regions into HDRs leaves some regions under-resolved. The graph allows repetitive regions to collapse, similar to the short-read alignments, while maintaining the sequence context. The detection of variants can nevertheless be hindered by the inability to project complex variants into the linear reference space, or simply under-aligned regions during graph construction. The high identity of SNPs between the different call sets shows that small variation can be called reliably from every type of input data, and are more trustworthy, whereas longer variation is more unreliable in its recallability.
The remaining differences in short variation can easily be explained as a result of different variant reporting. For example a short multi base pair variant called by an assembly based method may be reported as multiple SNPs in the short-read calls. This explains a large proportion of the SNPs that are not shared between all call sets and some of the small variation. Another example is the direct result of the described reference bias. As genome assemblies are contiguous sequences, resolved copy number changes result in PAV events insted of heterozygous variant calls. As a result of this small variant calls in the short-read based set can proxy for larger SVs. An advantage of assembly based variant detection methods is their ability to detect larger and more complex variation due to their longer continuity over short-reads. While the differences described above are a result of the properties of the query sequence in the comparison, another critical factor, in addition to the alignment method itself, comes from the available reference sequence data. The sequenced short-read alignments contain in principle the full range of sequence variation available but suffer from being mapped to a reference that does not contain all the sequences present in the query genome and mapping therefore misplaces short-reads, producing variant calls in inappropriate places, or lack the power to detect them entirely. In contrast the pairwise whole genome alignments use largely resolved chromosomal sequences to detect variation, but the assembly itself can lack sequence that would be represented in the short-reads. Furthermore, the tool *SyRI* aligns chromosome scale scaffolds to reference chromosomes, and thus is unable to call any variation that is present in unplaced contigs. The graph that has been used in this analysis also contains the unplaced contigs and therefore is able to utilize more genomic sequence than the pairwise alignment. Irrespective of the additional availability of sequence, the variants detected from the graph strongly depend on the quality of the graph structure and the ability of the algorithm to deal with highly complex regions. This results in either a reduced number of variants, or a concatenation into larger, under aligned blocks.

4.2 The sixRef proteome

The *auto-ant* annotation pipeline, which I developed for this thesis, could reliably annotate genomes. Especially the inclusion of *evidenceModeler* to combine multiple independent gene predictions improved the overall results. In the settings I opted for a higher specificity over more annotated features. The *augustus* annotation software has been a mainstay for ab-initio gene prediction and has been used in multiple other annotation pipelines, such as *BRAKER* (Hoff *et al.*, 2019). This novel pipeline increases the annotation speed of *augustus* by chunking the assemblies and enables the tool to run annotations of multiple assemblies in parallel. The annotations of different assemblies are then related to each other using an orthogroup assignment. The usefulness of the upfront TE annotation and its masking is proven by the very low number of TEs annotated in the final gene annotation, despite their presence in the genomes and the ability of *augustus* to annotate these. This is especially useful to deal with retrotransposons that have an RNA intermediate (Wicker *et al.*, 2007) and would otherwise be annotated by the RNA based annotation steps. The, in comparison, higher number of transcript related features in the RNA based *augustus* annotation supports the decision to rate the trustworthiness of this annotation higher than the other annotations. The high similarity of the six genome annotations in raw numbers and the orthogroup assignments, despite the highly variable mRNA evidence, further supports the robust performance of the *auto-ant* pipeline. Errors in the annotation mostly occur in the form of gene fusions that create overly long gene transcripts, but those events are rare. The gene annotations are coherent with the reference annotation. Most of the genes are orthologous to genes in the *araport11* annotation and conserved in their order, orientation and copy number. This, again, highlights the strong synteny of the assembled genomes and reference genome. The localization analysis of genes showed the same structural variation pattern as discovered in the *SyRI* analysis. The TE annotation using *EDTA* also showed the expected patterns of TE class distribution. Gypsy LTR TEs are more prevalent in pericentromeric regions, while copia LTR TEs more common in gene dense parts of the genome (Hufford *et al.*, 2021).
The pan-proteome follows the expected U-shaped distribution of *private*, *shell*, and *core* orthogroups that can be observed in all pan-genomic analysis. I can even observe the reminiscence of ancient genome duplications in the duplication patten of orthogroup copy numbers (Simillion *et al.*, 2002; del Pozo and Ramirez-Parra, 2015). In the reference free analysis I observed that *core* orthogroups were expanded more often than contracted. This behavior is an artifact of the *core* definition. In order to be considered as a *core* orthogroup, this group has to contain at least one gene from each annotation. As the majority of orthogroups are single copy orthogroups, the loss of a gene copy in a single accession would remove this orthogroup from the *core* set. In contrast the reference based orthogroup description contains more contracted orthogroups. This is another example of reference bias.
The *araport11* reference annotation (Cheng *et al.*, 2017) contains more genes than the more conservative *de-novo* annotation produced by *auto-ant*. As a result of this, the

additional genes in the manually curated reference annotation create the erroneous impression that the orthogroups are contracted. Despite this, I have been able to discover 148 orthogroups that are shared among all assembled accessions, and the outgroup, *A. arenosa*, but are not represented in the reference annotation. These genes are most likely core genes that are not represented in the reference annotation. The fact that 62.9% of them are located inside variable regions in the graph makes it very likely that their genomic sequences are not present in the reference assembly, further adding to the potential bias of the reference. While only 30.3% of them are supported by RNA transcription evidence, which is considerably lower than the average support of annotated genes (78.8%), at least these have a high confidence of being expressed genes that are part of the pan-proteome of *A. thaliana*, but are not represented in the known reference annotation. The remaining genes might be pseudogenes, or artifacts of the annotation approach, that happen to intersect with annotation of the outgroup, that has also been annotated using *augustus*. An in-depth analysis of these genes would be interesting, but has not been performed yet.

4.3 Graph genome

The constructed genome graph is able to represent the pan-genome of the six *A. thaliana* accessions and the *TAIR10* reference genome (Berardini *et al.*, 2015). The 79% graph *core* sequence in the reference genome corresponds well with the 77.2% of syntenic sequence in the reference that is shared among all genomes. The difference can easily be explained by the slightly different definitions. The syntenic sequence detected by *SyRI* contains internal variation, while the graph collapses repetitive regions. The good representation of sequence synteny in the graph is, in part, the result of the additional smoothing step in the *pggb* pipeline. This step increased linearity and alignment rate in the graph, as indicated by the decreased node degree. This drop is the result of the alignment of previously unaligned sequences of the graph and the splitting of over-connected components in the graph. This can also be observed in the decreased node size. Despite the best effort to resolve the graph, overly connected nodes and misalignments still prevail in the graph. This becomes very obvious in the *Kraken2* (Wood *et al.*, 2019) analysis of the non-reference sequences. 79.4% of the classified sequence was attributed to *A. thaliana*. This is a direct result of unaligned sequences that exist in the graph, but are represented by the reference genome. Based on the annotation of these regions it becomes obvious that a majority of them are either repetitive or belong to the mobilome. As graph construction has to be a trade-off between compression and usability the limit that I had to impose on repeat copies to collapse, as well as the synteny driven underlying alignment will have had an impact on the presence of unaligned mobilome sequences. In turn this has resulted in a more linear graph, where the pericentromeric regions are highly connected among the chromosomes, but the chromosome arms are mostly linear. The overall compression rate of sequence in the graph is comparable for all six *de-novo*

assemblies. Only the *TAIR10* reference genome exhibits a lower compression rate. This is most likely a result of the under-resolved repeat space in the reference that is being compressed in the *de-novo* assemblies.
The use of *panSV* enables a better resolution of the variation in the pan-genome over traditional reference based methods. When using this method we have to keep in mind that the representation of variation by *panSV* is very different compared to the traditional vcf format. This is a result of the core level based variant definition that does not require a variant to be anchored to a singular reference genome, but describes variation in the context of pan-genome frequencies. This can explain the differences in the comparison of variable regions detected by *panSV* and the results of the *TAIR10* (Berardini *et al.*, 2015) based *vg deconstruct* (Garrison *et al.*, 2018) results, that were both obtained from the same genome graph. Most variation reported by *panSV* is present at the highest *core* level, and intersects with the reference based variants. The strength of *panSV* becomes evident in the complex and nested regions of the graph. While *vg deconstruct* is not able to fully represent the nested variation, *panSV* resolves those complex regions and reports a multitude of variants that are nested within, where *vg deconstruct* only reports a single large variant. This is the reason for the higher fraction of SNPs called by *panSV*, and the lower fraction of *small variants*. An additional driver for the difference in detected variation are repetitive regions, due to the way they are represented in the graph. As *panSV* searches for regions with diverging *core* levels, it is unable to describe repeats that do not result in a change in the *core* level. Nevertheless, this new approach will enable us to access nested variation that has been hard to describe and might become an additional method to describe the complex variation in a more conceivable form.
Despite the shortcomings in the graph resolution I was able to detect parts of the mobilome in the variable regions of the graph using *panSV*. While the basic algorithm is very simplistic in its description of the graph, the reference free approach helps to deal with variation that tools like *vg deconstruct* struggle to represent. In highly repetitive regions it can be challenging to project variation into a linear coordinate system. This problem is entirely circumvented by my approach. Nevertheless it does not come without its own set of challenges. One of which is the interpretation of the results. While the concept of polarized, reference based variation is well understood and easy to grasp, the complexity of traversals and their nestedness can be hard to understand and even harder to visualize. This concept of pan-genomic variation will need time to be refined and get used to. Nonetheless I was able to successfully use this approach to detect the traces of insertion mechanisms of the mobilome and explain a subset of the non-standard orthogroups. TEs, genes and a combination of both is enriched in variable regions that are anchored by identical, thus repeated, nodes on both ends. This is a result of the insertion and breakpoint repair mechanism (Chatterjee and Walker, 2017). The high fraction of those regions that contain both, TEs and non-standard orthogroup genes, indicates that those genes might be dragged through the genome alongside TE.s A more in-depth analysis of the remaining bubbles could possibly reveal previously undescribed players of the mobilome.

4.4 Graph based short-read alignments

Alignments to graphs are one of the biggest challenges when working with genome graphs. Only a limited number of genome graph alignment algorithms habve been implemented at the moment, and most of them have been geared towards vcf-derived *vg* graphs. For this project I evaluated the performance of different algorithms on complex *pggb* graph (Garrison *et al.*, 2023). Two of the alignment algorithms are part of the *vg* toolkit (*vg map*, and *vg giraffe*) (Garrison *et al.*, 2018). I also used *graphAligner* (Rautiainen and Marschall, 2019), and a novel combination of *bwa mem* (Li, 2013) and *vg inject* to project linear alignments into the graph space. While *vg map*, and *vg giraffe* had a superior performance on linear, and vcf-derived graphs their memory consumption increased, and alignability decreased on complex graphs. Here they struggle with the highly complex regions of the graph that challenge their alignment algorithm and bloat the calculations they need to perform to a degree where alignments become infeasible. While *graphAligner* did not suffer from the same limitations, and was exceptionally fast, its implementation for long-reads made it unsuitable for short-read alignments and resulted in an inflated amount of covered sequence. A short-read implementation of the algorithm is in development, but has not been released at this point. This left me with the *vg inject* based approach, that by design had a constant performance on all suitable graphs, but suffered from its own set of limitations. The underlying idea has a very simplistic beauty in that it uses the well established alignment method to flat sequences and the positional relationship between flat genomes and paths in the graph to then inject the alignments into the graph. While the additional sequences present in the assembled genomes allow more reads to be aligned, the fact that the initial alignment target contains multiple copies of the same sequence unnecessarily slows down the alignment step. This can become infeasible with more, and larger genomes. In addition this method can only align reads to allele combinations represented in one of the genomes and crossovers are not possible. Still it is the best option to date to align short-reads to highly complex graphs.

In addition to the evaluation of alignment tools we can also observe the impact of further graph compression onto the alignment statistics. While the fraction of reads aligned by *vg giraffe* stays almost identical for the step from the *chromosome graph* to the *linear graph*, the amount of covered sequence decreases by almost 2 Mb when aligning to the *complex graph*. This is a result of the increased compression, especially in the pericentromeric regions.

The alignment of short-read sets from 840 accessions from the 1001 Genomes Project (1001 Genomes Consortium, 2016) demonstrates that the graph makes additional sequence available as additional mapping targets increase the number of alignable reads. This is directly reflected in the ability to cover more sequence in the graph than the length of the *TAIR10* reference genome, and align more reads compared to alignments to the *TAIR10* reference genome. This is especially visible in re-mapping of the three accessions that are part of the graph, and the 1001 Genomes Set. The amount of sequence cov-

ered by these alignments exceeds the length of their repective genome assembly. This is result of sequence that could not be assembled in their genome assembly, is reperesented by one of the other assemblies and thus becomes availabel for mapping. This demonstartes that multiple incomplete references in a graph can together represent more of their individual sequences. The ability to cover more sequence in the graph than the length of the reference genome is also a true biological signal of additional alleles from the novel assemblies, as well as a better representation of extensive copy number variations present in *A. thaliana* (Jaegle *et al.*, 2021). Nevertheless, a varying fraction of reads remained unaligned in each of the short-read sets. Only a small fraction of these reads could be assigned to *viridiplantae*. Instead a larger fraction has been classified as belonging to a *non-viridiplantea* clade and as such are most likely a result of samples not coming from sterily grown plants, which are naturally colonized by microbes. As *Kraken2* (Wood *et al.*, 2019) masks repetitive regions, the majority of unclassified reads most likely also originate from those. This is in line with the observation that the amount of unaligned genome sequence, by the k-mer based size estimation, is largely independent from the amount of unaligned reads. Nevertheless the sequences in the contaminated sets slightly bias genome size estimation analysis.
The analysis of the aligned, and estimated genome size reveals another important bias in science. While there is a significant correlation between the genome size and the admixture group, there is an even stronger correlation with the laboratory that sequenced the data set. Especially the Swedish accessions, which show the highest amount of covered sequence were mostly sequenced at a single sequencing center that sequenced few other accessions in the 1001 Genomes Project. It is very likely that the size differences are a bias that was introduced by different handling of the plants and material, or sequencing protocols by different experimenters (Stoler and Nekrutenko, 2021).
I already described the pan-genome of the six *de-novo* genome assemblies. By using the graph as a target for short-read alignment I can now use it to describe the pan-genome of the mapping population. A shift towards the higher pan-genome levels can be observed in the comparison of the two pan-genomes. Especially the shift from *private* sequence to *shell* sequence is noticeable. This shift is a result of the glass roof imposed by the graph, and another form of reference bias. As the sequence space in the graph is finite I can only describe sequences that are present in the graph, and thus the identification of novel sequences is impossible. This drives the shift away from *private* sequences in the mapping population as previously *private* sequences are in fact underrepresented in the assemblies. Another driver is the under alignment of the graph, that I described in the non-reference analysis. As this sequence is in fact more common than it is perceived in the graph based pan-genome it becomes a *shell*, or even a *core* sequence in the mapping based pan-genome. Nevertheless the fact that 82.2% of the core sequence is identical in both pan-genomes shows that we can already represent a large fraction of the *A. thaliana core* genome with just seven, well chosen, assemblies.
In the pan-proteome the same behavior as in the pan-genome analysis can be observed. Again the number of *core* orthogroups increases while the *shell*, and especially *private*

categories shrink. The shift in *shell*, and *private* graph sequence to the next (higher) category is a hint at underestimated alleles that appear to be rare in the graph, but in fact are more common in the larger population. This theory is backed by the per-accession analysis of the pan-proteome. The three accessions that are part of both, the assembled genomes, and the mapping population, have higher numbers of *private* orthogroups in the mapping population based analysis, compared to the other accessions in the mapping population. Especially the high number of *private* orthogroups in short-read mapping of the relict accession *AT6911* shows that we are still missing rare alleles of the wider population. Adding more diverse genomes to the graph structure will open up additional alleles. The pan-proteome analysis revealed another way the current *TAIR10* reference genome imposes a bias onto analysis performed with it. The set of 148 *core* orthogroups that contain a member form *A. arenosa* but have no member from the reference annotation, are also present in the majority of the mapping population, further underlining their status as common genes of *A. thaliana*. While this problem is not as severe as for example in maize (Hirsch *et al.*, 2014; Lu *et al.*, 2015; Hirsch *et al.*, 2016; Jin *et al.*, 2016), or in wheat (Bayer *et al.*, 2022), researchers in *A. thaliana* have already begun to choose alternative references to better represent the causal genetic components for their research question (Wójtowicz and Gieczewska, 2021). Here, genome graphs and pan-genomes can help us to better represent and understand the true genomic potential of a species. While we are already capable of expanding the core genome of *A. thaliana* with the addition of just six assemblies, it also means that we will need to add more diverse accessions to be able to represent the rare alleles of the population. Never the less the finite sequence space of a graph will never be able to represent the full library of sequences.

Variant detection is the kind of reference-based analysis that suffers the most from reference bias. Therefore the analysis of the graph based calls can help us to better understand how a genome graph changes, and reduces the reference bias in such analysis. Similar to the analysis of the variation stored in the graph itself, the distribution of variant types shifts from predominantly *SNPs*, and a few rare *small variants*, to an almost equal number of sites that are categorized as *SNPs* and *large variants*. The underrepresentation of larger variants in previous analysis is a result of their inability to detect them. In a graph these variants are represented and can be called. Such variants are either a representation of large non-reference alleles, or of an incomplete graph resolution. This is further supported by the localization of *large variants* in the genome. While *SNPs* are distributed along the chromosome, *small* and *large variants* are mostly found in the pericentromeric regions, where the *de-novo* assemblies were able to sequence more, and deeper into complex regions of the genome. The main difference between *SNPs* and *large variants* is their frequency within the mapping population. Sites that are classified as *SNPs* are shared by more samples, while *large variants* are mostly private to a single sample. Even though the sequence of the underlying PAV event might not be rare in the population, nested variation within these regions result in a multitude of low frequency variants that differ by only a few bases, and thus seem unique. This also artificially in-

flates the heterozygosity rate in the population. The potential causes of this kind of new reference bias will be discussed at the end of this section.
The comparison with the calls made by the 1001 Genomes Project (1001 Genomes Consortium, 2016), using the same short-read data reveals a reduction and shift in reference bias. While the number of variants detected by both methods are very similar, they only intersect for 70.1% of the sites per accession. This can partly be attributed to the fact that the graph allows to call larger variants, but also the difference in the post processing of the variant calls. The graph calls were subjected to a very basic quality filtering that kept most of the variants, while the 1001 Genomes Project calls have been extensively filtered and heterozygous calls have mostly been removed. This means that in reality the graph produced substantially fewer variant calls. This can also be seen in comparison with the re-analysis of the reads in the heterozygosity study by Jaegle et. al. (Jaegle *et al.*, 2021). Their initial calls resulted in 3.3 million SNPs. Which is almost twice the amount of total sites I called from the graph for all variant types, and three times the number of SNPs. Even though the majority of variants, called by the 1001 Genomes Project, intersect, or overlap with the variants called from the graph, not all variants could be recalled. Even though no variants were called at these positions, 90.9% of them were covered by reads in the graph. This means the reads aligned over the previously variable position matched perfectly. As such the previous variant was probably a result of the inclomplete sequence representation in the reference genome.
Multiple factors are responsible for the differences between the two call sets. They are either true biological signals as a result of the reduced reference bias, or a new bias that has been introduced by the graph and the way it has been constructed. First of all the better representation of the pan-genome sequence in the graph reduces the number of misplaced alignments that would otherwise result in variant calls caused by the incomplete representation of the reference genome. In addition small variants that have been called in the 1001 Genomes Project may actually proxy for larger variants, which could not be represented by the short-reads and the reference sequence, but are now resolved, and represented by the *de-novo* assemblies. The covered variants that could not be recalled probably belong to this category. In addition the graph itself also biases the variant calls. This bias, and its result will be discussed at the end of this section.
Beyond the simple comparison of intersecting variants, the rate of heterozygosity also tells an interesting story. As described before, the main contributors to heterozygosity in *A. thaliana* are extensive gene duplications that are not represented in the linear reference genome (Jaegle *et al.*, 2021). The genome graph enables us to better represent this set of variable sequences and therefore correctly place the corresponding reads. This results in an overall reduction of SNPs and heterozygous calls. I call just 44.4% of the SNPs have been called by Jaegle et.al. and only 4.6% of them were heteroygous per line, which is a massive reduction. Some of this is due to the fact that I used 840 accessions, instead of 1,057, but the main contributor to this reduction is the better sequence representation in the graph. Nevertheless the overall heterozygosity (14.2%) in the variants remains higher than expected in a selfing plant. The cause for this can be twofold. As we can observe

in the distribution of heterozygous calls in *AT6909*, most of them are located around the pericentromeric regions that are expected to have a higher mutation rate and therefore maintian more variants eben in selfing plants. Thus some of them are real heterozygous calls, but a second group are them ares probably the result of a new type of reference bias that is introduced by the graph representation of the sequence, which I will discuss next.

While the graph resolves some problems surrounding the incomplete sequence representation of the reference genome, it also introduces a new type of bias. In the graph, sequences with variable copy numbers are either compressed into subgraphs where the differences between copies are collapsed, or left unaligned as large PAVs in the graph. As a result of this we can observe a higher number of *small* and *large variants* in the variant calls. They either represent the under aligned fraction of the graph, or genuine, new sequences. In addition this also results in a inflation of heterozygous calls. While the diffent copies of a region are assembled and represented in the graph, collapsed sub graphs can make the placement of novel variation dificult and result in heterozygous calls in these sub graphs. In addition nested varition in larger variants also creates heterozygous calls, as they can not be represented in the reference based vcf file. This problem is especially prevalent in the mobileome, and can be observed in the comparison of mappings of the reference accession, *AT6909* with the *TAIR10* reference genome. It is clearly visible that the heterozygous variants coincide with regions of high TE density. As the pericentromeric regions have not been fully resolved the diverging TE copies easily cause heterozygous variant calls. The subset of heterozygous calls with two alternative alleles in this accession highlights that we have not fully resolved it in the *de-novo* assemblies. The exact degree by which this bias influences the variants calls needs to be determined, and addressed in the future.

Chapter 5

Conclusion & Outlook

In my work I have shown that building highly complex genome graphs from whole genome assemblies of *A. thaliana* is feasible. The additional sequence represented in the six *de-novo* assemblies is able to enrich the current reference genome. Even using the whole genome assemblies for pairwise comparisons allows me to minimize the detection bias of SNPs over structural variants. This bias has forced researchers to describe possibly less impactful SNPs. By annotating the new assemblies with my novel *auto-ant* annotation pipeline I was able to detect novel genes that are conserved in the six genomes, and in the larger mapping population of 840 accessions, but have not been part of the reference annotation. A functional analysis of these genes is required, but has not been performed yet. While this is an interesting result it once again highlights the problem of reference bias, which can be minimized by adding additional genome assemblies to enlarge the sequence and proteome space. Nevertheless the reference bias is just reduced and not eliminated. While the graph itself helps to place more reads than the linear reference genome, there are still reads that remain unmapped and can be classified as *viridiplantae*. Six additional genomes are not enough to represent the variation of the larger population, but are just a step in the right direction.
Future studies, like the next iteration of the 1001 Genomes project, will need to investigate the impact of a higher number of assemblies onto the representation of the pan-genome. While it is not an imminent problem yet, a useful cutoff for the inclusion of additional accessions into the graph will need to be defined. Otherwise a genome graph can become infinitely complex. A second problem will be a method to update graphs, as currently a graph will need to be completely reconstructed everytime a new accession is added. This re-construction will change the landscape of the graph and will have an impact onto the analysis performed on the graph. The currently largest problem in the field of genome graphs is the construction of those structures. Although I have been able to show that genome graphs can be constructed from whole genome alignments of assemblies, the current construction tools still suffer from our poor understanding of large scale variation and only recently started to evolve to tackle the unique problems of whole genome and multiple whole genome alignments (Minkin and Medvedev, 2020; Garrison *et al.*, 2023). As a result of this my graph still remains under-aligned. But even while my dissertation work was in progress, tools evolved from k-mer based approaches and

at that point barely usable *cactus* alignments (Paten *et al.*, 2011; Armstrong *et al.*, 2020) to the current *pggb* pipeline (Garrison *et al.*, 2023) and is still evolving. Not included in this work are recent developments in the alignment and smoothing step of the graph, that further improve the results.
Despite the problems in the graph construction itself, I was able to mitigate the impact of the reference bias using the graph. This reduction is not only true for sequence alignments to the graph reference, but especially for representation of variation in the graph. With the implementation of my *panSV* algorithm I could start to describe variation without the inherent reference bias and naturally describe nested variation. The easy discovery of mobile elements, their insertion mechanism, as well as the explanation of a set of mobile genes discovered from the six *de-novo* annotations is only the starting point of a long journey to open the treasure trove this new angle on structural variant detection might offer. Further work on this kind of data could easily reveal new classes of mobile elements and offer a deeper understanding of evolutionary processes based on nested variation. This method, only applicable to whole genome derived graphs, is a distinct advantage of those graphs over vcf-derived graphs, as I am able to easily describe the context of a variant in every input genome.
The second, and most obvious area of reference bias reduction are read mappings and the corresponding variant calls. Here the graph of six assemblies already reveals the extent of biased variant calls and heterozygosity induced by the old reference. New, graph based applications, such as in GWAS, will be able to find much clearer associations as the noise in the variant calls have been massively reduced (He *et al.*, 2023; Ebler *et al.*, 2020). In order to further refine this process there is a need for reliable mapping algorithms. The currently used algorithms provided by *vg* (Garrison *et al.*, 2018) are tailored towards simple vcf-derived graphs. Currently *vg giraffe* is able to cope with low complexity alignment derived graphs. As the *vg* toolset is under constant development and improvements to the alignment algorithms are released regularly this limitation of *vg giraffe* might become obsolete at some point. The current state of the injection based method I applied does not scale well with larger genomes and larger graphs. It is also limited in its ability to resolve variation. In order to fix this, I propose two additional computational steps that, while requiring the development of new methods, will make this easy method more applicable. To tackle the problem of scaling with an increasing number of input genomes, the genomes need to be compressed on a higher level than the resolved graph and these compressed fragments need to be added as paths to the graph. These paths then traverse through existing nodes in the graph and represent the main alleles that are too large for reads to bridge them. In this step a low complexity, consensus, version of paths in the graph will be created by omitting the smaller variants. These consensus paths can then serve as a template to create an alternative haplotype file to be used with the bwa mem toolkit (Li, 2013). This allows us to use the well validated alignment method and make additional alleles available without having to use complex graph alignment method, or large numbers of mostly identical genomes. The alignments to the consensus references will then be injected into the full graph, as shown in this

thesis. The second additional step will improve the alignments. It will locally realign the injected reads in the more complex graph. As the reads were aligned to less complex consensus paths this step is necessary to account for the variants that hav ebeen removed in the consensus sequences. In addition this will solve the problem of allele combination and is computationally less expensive than a full graph alignment. Sadly the implementation of this was outside of the scope of this work.
Even though the better representation of variable sequence in a genome graph reduces the reference bias, and reduces the number of variants called from short-reads, it also introduces a new bias. The shape of this bias depends on the degree of sequence compression in the graph. Graphs that compress most copies of a sequence into a subgraph will see a higher number of heterozygous SNPs, while graphs that reinforce synteny will result in a higher number of larger, heterozygous, variant calls. Therefore the optimal degree of compression will remain an option question, as well as the way to cope with calls that are clearly a result of the way sequence is represented in the graph, and are later projected into a linear reference space in the variant calling step?
While obvious challenges remain, the field of graph genomics is very active, with many bright minds involved. There is good reason to believe that in the coming years further significant improvements will be made.

Appendix A

Abbreviations & Glossary

Bubble	A structure in the graph topology that is being formed by variable sequence. It is defined as a closed subgraph with an upstream and downstream anchor *node* and a set of *nodes* in between them that represent variable sequence (Figure 1.1). A looser formulation of this concept are *snarls* (Paten *et al.*, 2018).
Core	Part of a *pan-genome* or *pan-proteome* that is shared by all, or a large fraction of the accessions that are part of it
Core level	A concept used by *panSV* to describe the sharedness of a *node* in the *pan-genome*. The core level is defined as the number of genomes that contain the sequence stored in this *node* and can differ from the number of *paths* that traverse through it, e.g. in the case of a duplication event.
DAG	Directed Acyclic Graph - A type of graph that prohibits loops in its structure to go back to previous *nodes*.
Edge	Connective feature of a genome graph that orders and connects the *nodes*.
GFA	Graphical Fragment Assembly - a file format to store graphs in a human readable form (GFA, 2022).
HDR	Highly Diverged Region - region in a genome that contains a multitude of smaller variants, above the average of the surrounding sequence. As defined by the pairwise variant detection tool SyRI (Goel *et al.*, 2019).
InDel	Insertion-Deletion variation events in a reference framework that induces the polarity of sequence being inserted, or deleted from it. This term is slowly being replaced by *PAV*.
Mobilome	Fraction of the genome that consists of mobile elements such as *TEs*.
MUM	Maximal Unique Match - largest possible unique alignment between sequences
Node	Element of a genome graph that stores the sequence.

OG	Orthogroup - Group of orthologous genes.
Path	Colored traversal through a set of consecutive *nodes*, and *edges* of a graph. A path represents longer sequences in the graph, such as the input genomes. By following it through the graph the full sequence can be recovered.
Pan-genome	Combined collection of genetic sequence of multiple individuals to represent a larger population.
Pan-proteome	Collection of genes, or transcripts from multiple individuals that represent an enriched collection and the frequency of their occurrence in a larger population.
PAV	Presence-Absence variation events. A type of genetic variation where sequence has been gained or lost.
Private	Part of a *pan-genome* or *pan-proteome* that is shared by no other, or very few other individuals.
Shell	Part of a *pan-genome* or *pan-proteome* that is niether *core*, nor *private*.
Snarl	A structure in the graph topology that represents variable sequence in the graph. In contrast to a *bubble* this structure does not have to be a closed subgraph with up- and downstream anchors, but can have additional connections into it, or lack one of the anchors (Paten *et al.*, 2018).
SNP	Single-Nucleotide-Polymorphism - Variation between two DNA sequences where a single base is replaced by another single base.
Superbubble	A large substructure of the graph. It follows the definition of a *bubble*, but contains at least one smaller *bubble* inside.
SV	Structural Variation - Large sequence variation that alters the structure of the genome, for example PAVs, duplications, or translocations.
TE	Transposable Element - Mobile elements in the DNA sequence that are able to replicate themselves and insert into the genome.
Traversal	A collection of consecutive *nodes*, and *edges* in a graph that has a defined start and end *node*.

Appendix B

Supplementary

B.1 Supplementary Figures

Figure B.1: SixRef coordinates - Geographic locations of the six accessions chosen for assembly.

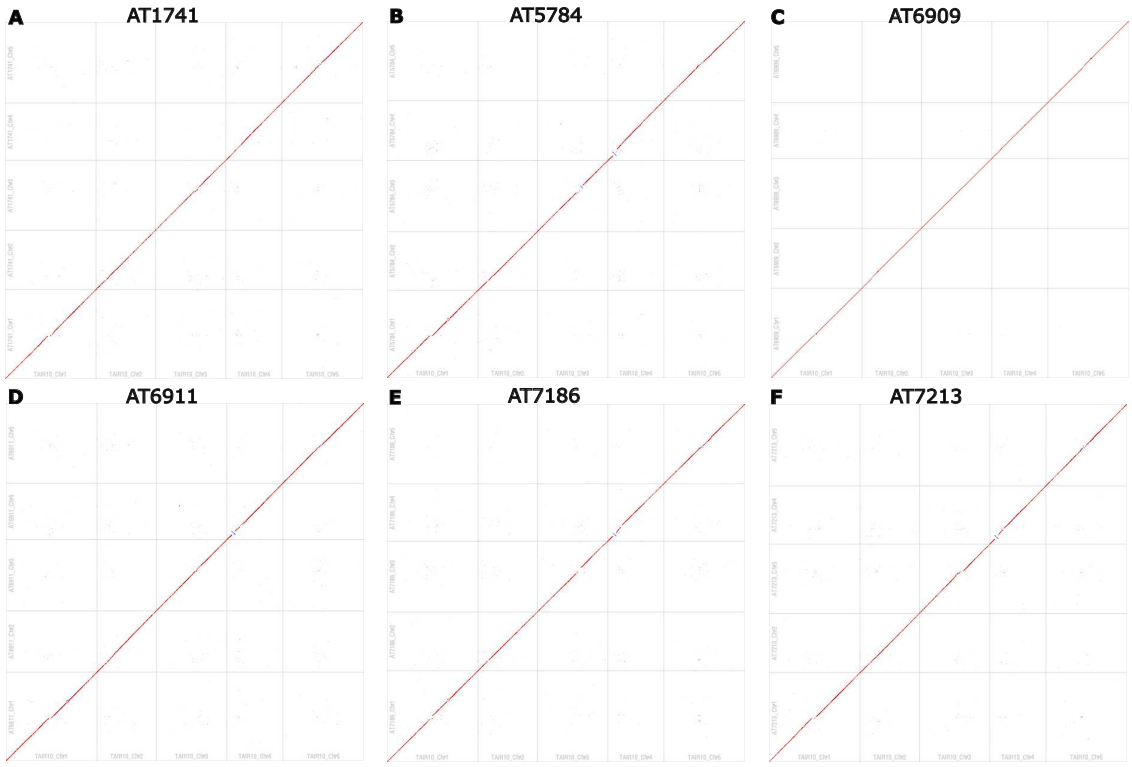

Figure B.2: Assembly dot-plots - Synteny representation of the six *de-novo* assemblies, based on *minimap2* alignments with the *TAIR10* reference genome to show the high level of synteny and reveal large scale inversions. Breakpoints are mostly located in the pericentromeric regions, together with a set of inversions.

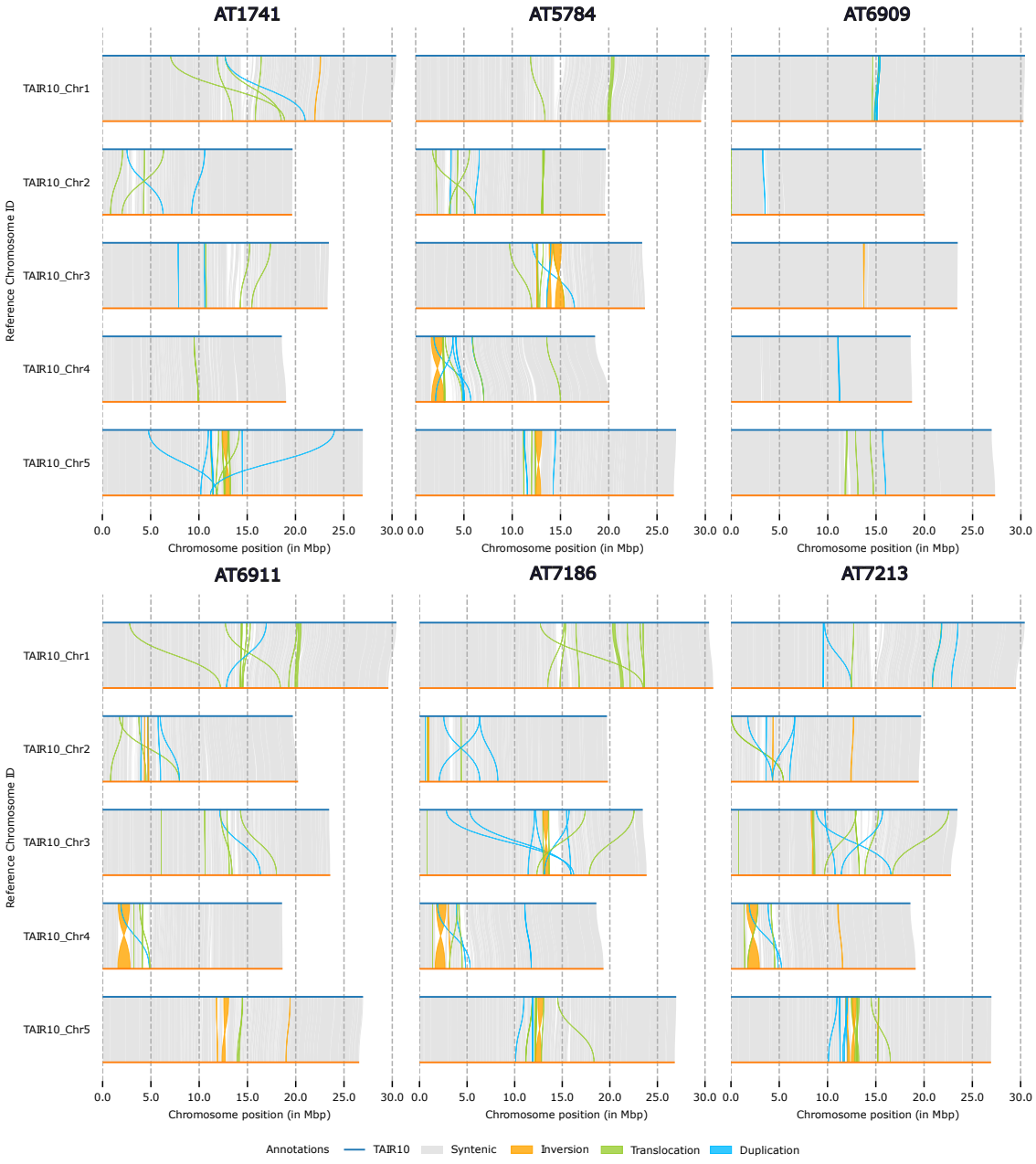

Figure B.3: *SyRI* rearrangements - Large scale structural rearrangements detected by *SyRI*. The majority of the chromosomes are highly collinear, and differences can mostly be observed in the pericentromeric regions.

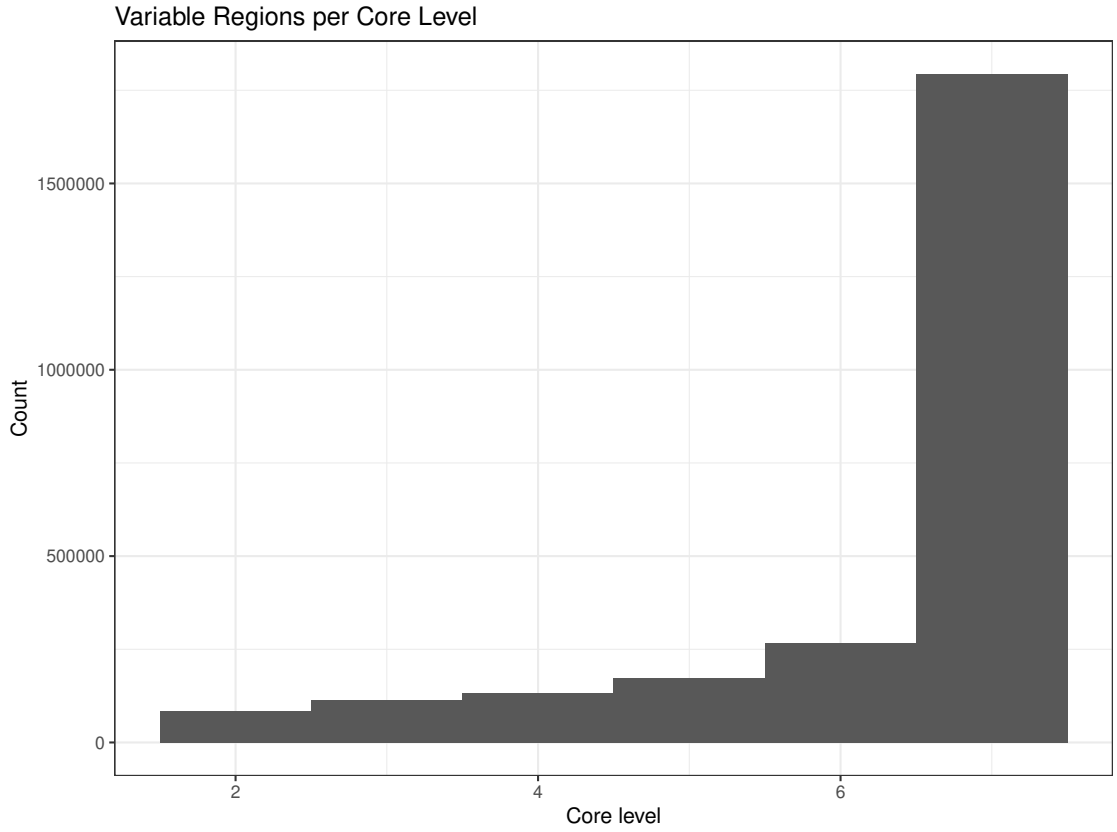

Figure B.4: *panSV* bubbles - Distribution of core levels of variable regions detected by *panSV*. Most of the regions have core level 7 and only a smaller fraction is nested within them.

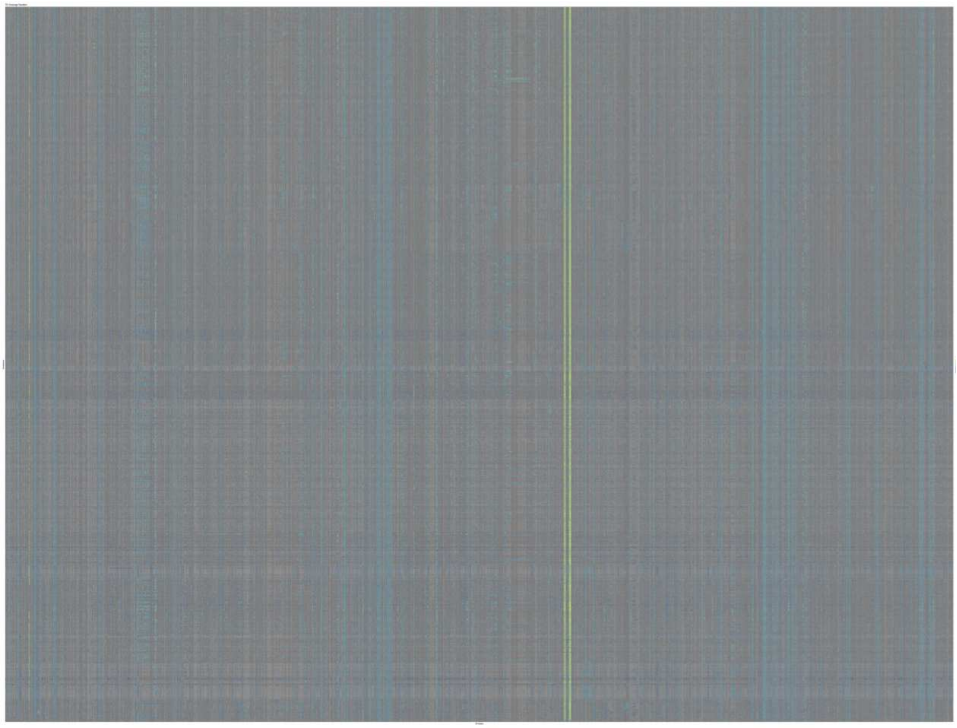

Figure B.5: TE node coverage - Heat map of the coverage of nodes (>50 bp) that are annotated as containing a TE in at least one accession of the graph.

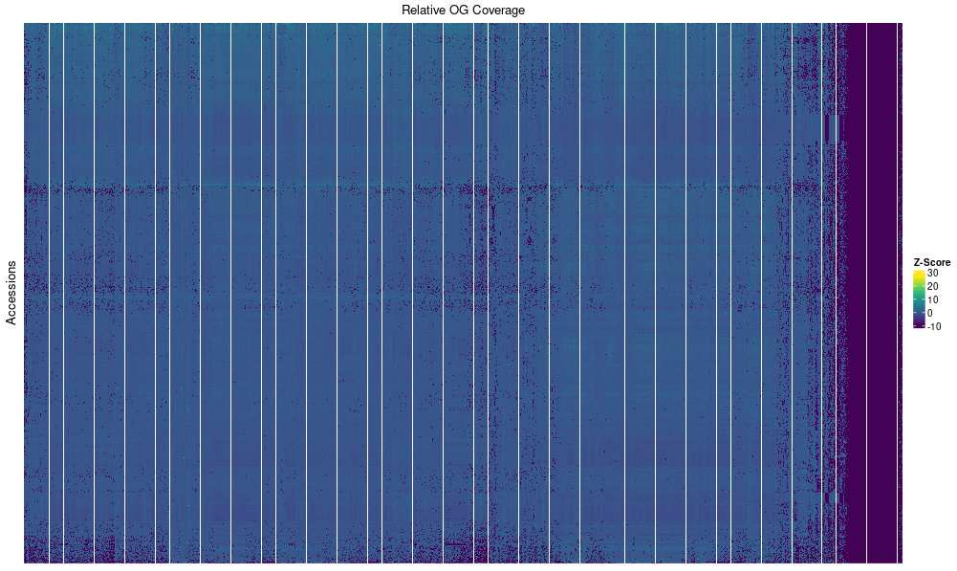

Figure B.6: Orthogroup Z-Score - Z-Score matrix of the estimated copy numbers of orthogroups, based on the number coverage of the accessions mapped to the graph.

B.2 Supplementary Tables

Table B.1: Reference based variant calls - Number and size of reference based variant calls, of the sixRef accessions, in comparison to the *TAIR10* reference genome, made with three different approaches. Short-read based variant calls made as part of the 1001 Genomes Project, pairwise alignment based variants called using *SyRI* from the *de-novo* chromosome scaffolds aligned to the *TAIR10* reference genome, graph based variants, extracted from the graph using *vg deconstruct*. The variants were classified as *SNPs* (one base pair replaced by another base pair), *small variants* (<50bp), and *large variants* (≥ 50bp).

	# Variants	% Variants	Size [Mb]	avg. size	median size
Short-read based variants					
Total	1,570,148	100	1.7	1.1	1
SNP	1,443,401	91.9	1.4	1	1
small variant	126,747	8.1	0.2	1.8	3
large variant	0	0	0	0	0
Pairwise alignment based variants					
Total	2,589,009	100	8.2	3.2	1
SNP	1,963,863	75.9	2	1	1
small variant	609,648	23.6	1.5	2.5	1
large variant	15,498	0.6	4.8	306.7	1
Graph based variants					
Total	1,869,605	1	88.6	47.4	1
SNP	1,272,633	68.1	1.3	1	1
small variant	539,644	0.28.9	2.2	4	2
large variant	57,328	3.1	85.2	1486.5	154

Table B.2: Variant call overlap - Variants that did not intersect with all other sets were overlapped with the remaining variants of the sets. Variants were classified as *small* (<50bp) and *large* (≥ 50bp). *SNPs* were excluded.

	small variants		large variants		
	# variants	%Intersection	# variants	% Intersection	Intersecting Sets
1001G	9,873	12.3	-	-	*SyRI*
1001G	44,293	55.4	-	-	*vg deconstruct*
SyRi	27,290	4.9	4,849	31.3	short-reads
SyRi	110,919	19.7	6,904	44.6	*vg deconstruct*
graph	78,855	16	11,465	20	short-reads
graph	112,561	22.8	11,807	20.6	*SyR*

Table B.3: *auto-ant* sensitivity - Features of each annotation intersecting with the annotated features in the *araport11* reference annotation. Some features were removed from the table as they did not intersect with any of the annotations. Those features, and their occurences in *araport11* were: lnc_RNA (2455), miRNA (427), miRNA_primary_transcript (325), pseudogenic_tRNA (27), rRNA (15), snoRNA (287), snRNA (82), transcript_region (726), tRNA (689), antisense_RNA (91)

	araport11	*BUSCO retrained augustus*		*LiftOff based augustus*		RNA-Seq supported augustus		*SNAP*		*cufflinks assembly*		combined evidenceModeler	
Feature	Feature count	Feature count	Sensitivity in *araport11*	Feature count	Sensitivity in *araport11*	Feature count	Sensitivity in *araport11*	Feature count	Sensitivity in *araport11*	Feature count	Sensitivity in *araport11*	Feature count	Sensitivity in *araport11*
CDS	286355	201840	0.7049	252047	0.8802	250829	0.8759	206895	0.7225	175032	0.6112	248384	0.8674
exon	200542	78245	0.3902	104870	0.5229	98845	0.4929	78428	0.3911	87987	0.4387	92489	0.4612
gene	33246	466	0.014	438	0.0132	298	0.009	586	0.0176	3	0.00009	652	0.0196
protein	48359	16652	0.3443	4897	0.1013	4963	0.1026	5308	0.1098	0	0	27729	0.5734
three_prime_UTR	41127	489	0.0119	956	0.0232	869	0.0211	511	0.0124	793	0.0193	627	0.0152
five_prime_UTR	46895	283	0.006	1406	0.03	1029	0.0219	301	0.0064	1010	0.0215	388	0.0083
mRNA	52141	0	0	556	0.0107	355	0.0068	778	0.0149	3	0.00005	919	0.0176
ncRNA	286	0	0	0	0	0	0	0	0	0	0	2	0.007
pseudogene	952	4	0.0042	3	0.0032	1	0.0011	11	0.0116	0	0	12	0.0126
pseudogenic_exon	2058	141	0.0685	178	0.0865	328	0.1594	220	0.1069	261	0.1268	194	0.0943
pseudogenic_transcript	1100	4	0.0036	3	0.0027	1	0.0009	11	0.01	0	0	12	0.0109
transposable_element	31189	10	0.0003	4	0.0001	5	0.0002	5	0.0002	0	0	8	0.0003
transposable_element_gene	3901	72	0.0185	14	0.0036	8	0.0021	155	0.0397	0	0	84	0.0215
transposon_fragment	34856	10	0.0003	4	0.0001	5	0.0001	5	0.0001	0	0	8	0.0002
uORF	111	2	0.018	6	0.0541	1	0.009	7	0.0631	1	0.009	0	0
antisense_lncRNA	1424	0	0	1	0.0007	0	0	1	0.0007	0	0	0	0

Table B.4: RNA-Seq reads - Raw count and fraction of RNA sequencing reads for each accession. The fraction always refers to the total number of raw reads.

		AT1741	AT5784	AT6909	AT6911	AT7186	AT7213
Count	Raw Reads	301,393,206	503,865,660	312,783,846	384,735,390	284,003,720	244,454,716
Fraction	Flower	22.5	11	46.6	47.8	24.7	18.4
	Leaf	22.8	12.4	19.4	15.1	15.1	31.
	Roo	27.4	61.9	19.4	28.2	25.8	32.2
	Seedling	27.4	14.7	14.5	9	34.4	17.7
	Trimmed	88.3	94.5	95.3	91.6	91.7	92.6
	Mapped	45.3	37.8	64.2	36	51.5	58

Table B.5: Non-reference *SyRI* intersection - Intersection of non-reference sequences with calls made by *SyRI*. Multiple events could be contained in one interval. I destinguished regions that were classified as not-aligned by *SyRI* and all other variants $\geq$ 50 bp detected by *SyRI*. The fraction relates to the total number of large non-reference sequences of the accession.

Accession	*SyRI* unaligned non-ref Regions	*SyRI* events	% non-ref Regions	*SyRI* SVs non-ref Regions	*SyRI* events	% non-ref Regions
AT1741	268	525	3.9	354	668	5.2
AT5784	313	586	4	478	782	6.1
AT6909	30	36	4.4	21	62	3.1
AT6911	365	682	3.9	503	845	5.4
AT7186	310	537	4.1	438	698	5.8
AT7213	303	588	4.1	421	748	5.60

Table B.6: Graph mapping test graphs - Statistics of the graphs used in the graph mapping evaluation. For each graph the sequence length, the sequence based multiple of the *TAIR10* reference genome, the level of compression compared to the combined input genomes are shown. In addition the Number of nodes, and edges, as well as the edge number divided by node number.

	Seq. Length [Mb]	Ref Fraction	Comp. Level	# Nodes	# Edges	Node Degree
Flat graph	119.7	1	0	3,829,320	3,829,313	1
VCF graph	220.4	1.8	25.76	5,601,134	8,307,979	1.5
Chrom graph	197.5	1.7	23.08	6,247,306	8,491,456	1.4
Linear graph	195.5	1.6	22.86	6,282,045	8,540,670	1.4
Complex graph	168.80	1.4	19.73	6,694,806	9,147,110	1.4

Table B.7: Graph mapping statistics - Statistics of mappings to the different test graphs using the prepared graphs with increasing complexity. Not all mapping algorithms could be used on all graphs. Either due to the method (*vg inject*, *bwa mem*), or the excessive run time and resource consumption (*vg map*).

		Mapped reads [%]	Covered graph sequence [%]	Covered bases [Mb]
Flat graph	*bwa mem*	96.16	95.42	114.2
	vg map	96.01	95.4	114.2
	vg giraffe	96.1	94.9	113.6
	graphAligner	56.79	97.1	116.2
	vg inject	96.16	95.42	114.2
VCF graph	*bwa mem*	-	-	-
	vg map	96.51	61.98	136.6
	vg giraffe	97.46	62.01	136.7
	graphAligner	44.44	80.39	177.2
	vg inject	-	-	-
Chrom graph	*bwa mem*	-	-	-
	vg map	56.24	30.49	60.2
	vg giraffe	97.69	68.27	134.9
	graphAligner	43.13	77.88	153.9
	vg inject	-	-	-
Linear graph	*bwa mem*	-	-	-
	vg map	-	-	-
	vg giraffe	97.67	68.5	133.9
	graphAligner	43.06	78.06	152.6
	vg inject	97.11	72.36	141.5
Complex graph	*bwa mem*	-	-	-
	vg map	-	-	-
	vg giraffe	73.37	42.27	71.3
	graphAligner	41.01	81.81	138.1
	vg inject	97.12	75.4	127.2

Table B.8: Mapping population - List of all accessions and their admixture group, that are part of the mapping population.

ID	admixture group	ID	admixture group	ID	admixture group	ID	admixture group	ID	admixture group	ID	admixture group	ID	admixture group	ID	admixture group	ID	admixture group
10022	admixed	9824	admixed	5907	central_europe	9772	central_europe	6909	germany	9646	italy_balkan_caucasus	6115	south_sweden	9841	spain	88	western_europe
1313	admixed	9830	admixed	5921	central_europe	9774	central_europe	6915	germany	9647	italy_balkan_caucasus	6118	south_sweden	9843	spain	9298	western_europe
1317	admixed	9831	admixed	5950	central_europe	9775	central_europe	6922	germany	9648	italy_balkan_caucasus	6119	south_sweden	9844	spain	9312	western_europe
2016	admixed	9835	admixed	5984	central_europe	9776	central_europe	6940	germany	9649	italy_balkan_caucasus	6122	south_sweden	9845	spain	9314	western_europe
2202	admixed	9838	admixed	5993	central_europe	9777	central_europe	6945	germany	9651	italy_balkan_caucasus	6123	south_sweden	9846	spain	932	western_europe
2276	admixed	9839	admixed	6008	central_europe	9778	central_europe	6982	germany	9653	italy_balkan_caucasus	6124	south_sweden	9848	spain	9503	western_europe
2317	admixed	9849	admixed	6296	central_europe	9779	central_europe	6997	germany	9655	italy_balkan_caucasus	6134	south_sweden	9850	spain	9517	western_europe
265	admixed	9858	admixed	6390	central_europe	9780	central_europe	7000	germany	9656	italy_balkan_caucasus	6141	south_sweden	9852	spain	9569	western_europe
5151	admixed	9877	admixed	6396	central_europe	9781	central_europe	7002	germany	9657	italy_balkan_caucasus	6149	south_sweden	9855	spain	9571	western_europe
5165	admixed	9881	admixed	6424	central_europe	9782	central_europe	7008	germany	9658	italy_balkan_caucasus	6191	south_sweden	9856	spain	9578	western_europe
5720	admixed	9890	admixed	6445	central_europe	9784	central_europe	7014	germany	9659	italy_balkan_caucasus	6192	south_sweden	9857	spain	9585	western_europe
5757	admixed	9892	admixed	6903	central_europe	9785	central_europe	7031	germany	9660	italy_balkan_caucasus	6203	south_sweden	9859	spain	9590	western_europe
5768	admixed	9894	admixed	6919	central_europe	9786	central_europe	7033	germany	9661	italy_balkan_caucasus	6413	south_sweden	9860	spain	9599	western_europe
5772	admixed	9897	admixed	6951	central_europe	9787	central_europe	7036	germany	9697	italy_balkan_caucasus	6974	south_sweden	9861	spain	9809	western_europe
5784	admixed	9906	admixed	6956	central_europe	9788	central_europe	7062	germany	9698	italy_balkan_caucasus	7013	south_sweden	9864	spain	9823	western_europe
5811	admixed	9924	admixed	6957	central_europe	9789	central_europe	7094	germany	9699	italy_balkan_caucasus	7164	south_sweden	9866	spain	9826	western_europe
5822	admixed	9932	admixed	6975	central_europe	9790	central_europe	7102	germany	9700	italy_balkan_caucasus	7343	south_sweden	9867	spain	9827	western_europe
6042	admixed	9933	admixed	6976	central_europe	9791	central_europe	7117	germany	9701	italy_balkan_caucasus	7346	south_sweden	9868	spain	9828	western_europe
6434	admixed	14312	asia	6979	central_europe	9792	central_europe	7119	germany	9703	italy_balkan_caucasus	7354	south_sweden	9870	spain	9847	western_europe
6830	admixed	14313	asia	6984	central_europe	9793	central_europe	7125	germany	9704	italy_balkan_caucasus	8234	south_sweden	9873	spain	9851	western_europe
6898	admixed	14314	asia	7025	central_europe	9794	central_europe	7133	germany	9705	italy_balkan_caucasus	8237	south_sweden	9874	spain	9853	western_europe
6907	admixed	14315	asia	7103	central_europe	9795	central_europe	7143	germany	9706	italy_balkan_caucasus	8307	south_sweden	9876	spain	9854	western_europe
6920	admixed	14318	asia	7106	central_europe	9796	central_europe	7160	germany	9707	italy_balkan_caucasus	8326	south_sweden	9878	spain	9862	western_europe
6932	admixed	14319	asia	7120	central_europe	9797	central_europe	7161	germany	9708	italy_balkan_caucasus	8426	south_sweden	9880	spain	9875	western_europe
6981	admixed	15560	asia	7158	central_europe	9798	central_europe	7162	germany	9709	italy_balkan_caucasus	8427	south_sweden	9882	spain	9908	western_europe
6987	admixed	18694	asia	7177	central_europe	9799	central_europe	7165	germany	9710	italy_balkan_caucasus	9339	south_sweden	9883	spain	9909	western_europe
6990	admixed	18696	asia	7203	central_europe	9800	central_europe	7192	germany	9711	italy_balkan_caucasus	9353	south_sweden	9885	spain	9910	western_europe
6992	admixed	1890	asia	7207	central_europe	9801	central_europe	7199	germany	9712	italy_balkan_caucasus	9369	south_sweden	9886	spain	9911	western_europe
7003	admixed	6929	asia	7223	central_europe	9802	central_europe	7202	germany	9713	italy_balkan_caucasus	9370	south_sweden	9888	spain	9912	western_europe
7028	admixed	6931	asia	7236	central_europe	9803	central_europe	7208	germany	9714	italy_balkan_caucasus	9380	south_sweden	9891	spain	9917	western_europe
7061	admixed	6938	asia	7296	central_europe	9804	central_europe	7231	germany	9716	italy_balkan_caucasus	9392	south_sweden	9895	spain	9918	western_europe
7072	admixed	6963	asia	7298	central_europe	9805	central_europe	7244	germany	9717	italy_balkan_caucasus	9413	south_sweden	9898	spain	9921	western_europe
7096	admixed	7183	asia	7333	central_europe	9806	central_europe	7248	germany	9718	italy_balkan_caucasus	9436	south_sweden	9899	spain	9925	western_europe
7126	admixed	7323	asia	7347	central_europe	9807	central_europe	7250	germany	9719	italy_balkan_caucasus	991	south_sweden	9900	spain	9926	western_europe
7127	admixed	763	asia	7349	central_europe	9808	central_europe	7255	germany	9720	italy_balkan_caucasus	6933	spain	9901	spain	9927	western_europe
7138	admixed	765	asia	7350	central_europe	9810	central_europe	7258	germany	9721	italy_balkan_caucasus	6961	spain	9902	spain	9928	western_europe
7147	admixed	766	asia	7372	central_europe	9811	central_europe	7276	germany	9722	italy_balkan_caucasus	6970	spain	9903	spain	9929	western_europe
7163	admixed	768	asia	7411	central_europe	9812	central_europe	7280	germany	9723	italy_balkan_caucasus	6971	spain	9904	spain	9935	western_europe
7169	admixed	772	asia	7424	central_europe	9813	central_europe	7282	germany	9725	italy_balkan_caucasus	7081	spain	108	western_europe	9937	western_europe
7181	admixed	8354	asia	7427	central_europe	9814	central_europe	7287	germany	9726	italy_balkan_caucasus	7327	spain	139	western_europe	9938	western_europe
7209	admixed	8424	asia	7460	central_europe	9815	central_europe	7337	germany	9749	italy_balkan_caucasus	8264	spain	159	western_europe		
7218	admixed	9125	asia	7520	central_europe	9816	central_europe	7356	germany	9755	italy_balkan_caucasus	8357	spain	350	western_europe		
7268	admixed	9128	asia	7521	central_europe	9914	central_europe	7358	germany	9757	italy_balkan_caucasus	9507	spain	351	western_europe		
7305	admixed	9130	asia	8235	central_europe	9915	central_europe	7359	germany	9759	italy_balkan_caucasus	9509	spain	4779	western_europe		
7314	admixed	9131	asia	8236	central_europe	9930	central_europe	7377	germany	1254	north_sweden	9510	spain	4807	western_europe		
7316	admixed	9133	asia	8284	central_europe	1612	germany	7416	germany	1257	north_sweden	9511	spain	4826	western_europe		
7319	admixed	9134	asia	8285	central_europe	1622	germany	7419	germany	1552	north_sweden	9512	spain	4840	western_europe		
7342	admixed	9607	asia	8290	central_europe	1651	germany	742	germany	5860	north_sweden	9514	spain	4884	western_europe		
7344	admixed	9608	asia	8311	central_europe	1652	germany	7461	germany	6010	north_sweden	9515	spain	4900	western_europe		
7353	admixed	9609	asia	8365	central_europe	1676	germany	7475	germany	6011	north_sweden	9518	spain	4958	western_europe		
7378	admixed	9610	asia	8386	central_europe	1684	germany	7515	germany	6153	north_sweden	9519	spain	5023	western_europe		
7382	admixed	9611	asia	8419	central_europe	1757	germany	7523	germany	6214	north_sweden	9520	spain	5210	western_europe		
7384	admixed	9612	asia	9644	central_europe	1793	germany	7525	germany	6221	north_sweden	9521	spain	5236	western_europe		
7394	admixed	9615	asia	9645	central_europe	1797	germany	7530	germany	6235	north_sweden	9522	spain	5249	western_europe		
7404	admixed	9616	asia	9664	central_europe	1819	germany	7566	germany	6237	north_sweden	9524	spain	5253	western_europe		
7418	admixed	9617	asia	9665	central_europe	1820	germany	7568	germany	6969	north_sweden	9525	spain	5276	western_europe		
7430	admixed	9619	asia	9666	central_europe	1829	germany	7717	germany	8227	north_sweden	9526	spain	5349	western_europe		
7471	admixed	9620	asia	9667	central_europe	1834	germany	7757	germany	8351	north_sweden	9528	spain	5353	western_europe		
7477	admixed	9621	asia	9668	central_europe	1851	germany	7767	germany	8376	north_sweden	9531	spain	5577	western_europe		
7514	admixed	9624	asia	9669	central_europe	1852	germany	7917	germany	9332	north_sweden	9532	spain	5644	western_europe		
8343	admixed	9625	asia	9670	central_europe	1853	germany	7947	germany	9363	north_sweden	9534	spain	5717	western_europe		
8366	admixed	9626	asia	9671	central_europe	1872	germany	801	germany	6911	relict	9535	spain	5726	western_europe		
9058	admixed	9627	asia	9672	central_europe	1925	germany	8037	germany	9533	relict	9537	spain	5741	western_europe		
9336	admixed	9628	asia	9673	central_europe	1954	germany	8057	germany	9542	relict	9539	spain	5779	western_europe		
9343	admixed	9629	asia	9676	central_europe	2017	germany	8077	germany	9543	relict	9540	spain	630	western_europe		
9381	admixed	9630	asia	9677	central_europe	2031	germany	8132	germany	9545	relict	9541	spain	6680	western_europe		
9383	admixed	9631	asia	9678	central_europe	2053	germany	8171	germany	9549	relict	9544	spain	6897	western_europe		
9506	admixed	9632	asia	9679	central_europe	2057	germany	8233	germany	9550	relict	9546	spain	6904	western_europe		
9508	admixed	9633	asia	9680	central_europe	2081	germany	8238	germany	9554	relict	9547	spain	6908	western_europe		
9513	admixed	9634	asia	9681	central_europe	2108	germany	8239	germany	9555	relict	9553	spain	6923	western_europe		
9523	admixed	9635	asia	9682	central_europe	2171	germany	8246	germany	9574	relict	9556	spain	6924	western_europe		
9527	admixed	9636	asia	9683	central_europe	2191	germany	8312	germany	9583	relict	9557	spain	6926	western_europe		
9529	admixed	9637	asia	9684	central_europe	2212	germany	8420	germany	9598	relict	9560	spain	6943	western_europe		
9530	admixed	9638	asia	9685	central_europe	2239	germany	8464	germany	9600	relict	9562	spain	6944	western_europe		
9536	admixed	9639	asia	9686	central_europe	2240	germany	8483	germany	9606	relict	9564	spain	6958	western_europe		
9551	admixed	9640	asia	9687	central_europe	2278	germany	854	germany	9832	relict	9567	spain	6959	western_europe		
9552	admixed	9641	asia	9689	central_europe	2285	germany	867	germany	9837	relict	9568	spain	6960	western_europe		
9558	admixed	9642	asia	9690	central_europe	2286	germany	868	germany	9869	relict	9573	spain	6966	western_europe		
9559	admixed	9643	asia	9692	central_europe	484	germany	870	germany	9871	relict	9577	spain	6967	western_europe		
9561	admixed	9737	asia	9693	central_europe	504	germany	9027	germany	9879	relict	9582	spain	6986	western_europe		
9565	admixed	9745	asia	9694	central_europe	5104	germany	915	germany	9887	relict	9584	spain	6989	western_europe		
9576	admixed	9758	asia	9695	central_europe	531	germany	9920	germany	9905	relict	9586	spain	7026	western_europe		
9579	admixed	9766	asia	9696	central_europe	544	germany	7068	italy_balkan_caucasus	1006	south_sweden	9587	spain	7064	western_europe		
9581	admixed	10020	central_europe	9727	central_europe	546	germany	7077	italy_balkan_caucasus	1061	south_sweden	9588	spain	7071	western_europe		
9591	admixed	10027	central_europe	9728	central_europe	5486	germany	9067	italy_balkan_caucasus	1062	south_sweden	9589	spain	7092	western_europe		
9592	admixed	15591	central_europe	9729	central_europe	5748	germany	9085	italy_balkan_caucasus	1063	south_sweden	9593	spain	7107	western_europe		
9595	admixed	15592	central_europe	9730	central_europe	5800	germany	9091	italy_balkan_caucasus	1158	south_sweden	9594	spain	7109	western_europe		
9596	admixed	15593	central_europe	9731	central_europe	6252	germany	9099	italy_balkan_caucasus	1166	south_sweden	9597	spain	7130	western_europe		
9622	admixed	19949	central_europe	9732	central_europe	6739	germany	9100	italy_balkan_caucasus	5836	south_sweden	9601	spain	7217	western_europe		
9663	admixed	19950	central_europe	9733	central_europe	6740	germany	9102	italy_balkan_caucasus	5865	south_sweden	9602	spain	7306	western_europe		
9691	admixed	19951	central_europe	9735	central_europe	6744	germany	9103	italy_balkan_caucasus	6022	south_sweden	9817	spain	7307	western_europe		
9736	admixed	403	central_europe	9739	central_europe	6749	germany	9104	italy_balkan_caucasus	6077	south_sweden	9819	spain	7320	western_europe		
9738	admixed	410	central_europe	9741	central_europe	6750	germany	9105	italy_balkan_caucasus	6087	south_sweden	9820	spain	7332	western_europe		
9743	admixed	424	central_europe	9747	central_europe	680	germany	9106	italy_balkan_caucasus	6091	south_sweden	9821	spain	7383	western_europe		
9744	admixed	428	central_europe	9756	central_europe	6805	germany	9111	italy_balkan_caucasus	6094	south_sweden	9822	spain	7387	western_europe		
9748	admixed	430	central_europe	9761	central_europe	6806	germany	9113	italy_balkan_caucasus	6095	south_sweden	9825	spain	8214	western_europe		
9754	admixed	5837	central_europe	9768	central_europe	681	germany	9114	italy_balkan_caucasus	6096	south_sweden	9833	spain	8243	western_europe		
9762	admixed	5874	central_europe	9769	central_europe	6814	germany	9115	italy_balkan_caucasus	6101	south_sweden	9834	spain	8244	western_europe		
9764	admixed	5890	central_europe	9770	central_europe	685	germany	9121	italy_balkan_caucasus	6105	south_sweden	9836	spain	8297	western_europe		
9783	admixed	5893	central_europe	9771	central_europe	687	germany	9613	italy_balkan_caucasus	6109	south_sweden	9840	spain	8337	western_europe		

Table B.9: Graph coverage - Amount of sequence covered by accessions mapped to the graph, separated by their admixture group, or the producing lab. The sequence is represented as mega bases.

		Mean Length	Median Length	Shortest Sequence	Longest Sequence	Number of Accessions
Admixture Group	admixed	119.9	120.2	110.2	132.2	118
	asia	120.5	120	111.1	135.3	65
	central_europe	121.6	121.3	112	133.4	162
	germany	120.6	120.4	110.2	132.1	137
	italy_balkan_caucasus	120	120.1	109.1	127.4	62
	north_sweden	124.9	125.2	118.7	129.4	17
	relict	122.5	123.4	113.1	129.8	21
	south_sweden	124.7	125.4	112.3	129.6	52
	spain	121.7	121.8	112.4	136.7	104
	western_europe	120.8	120.6	106.4	134.6	102
Producing Lab	Monsanto	120.8	121.1	106.4	127.4	482
	Salk	120.2	118.9	109.1	136.7	202
	MPI	120.9	120.7	113.3	129.3	77
	GMI	125.4	125.3	118.7	132.9	90

Table B.10: Pan-genome comparison -Comparison of the pan genomes of the six accessions in the graph with the pan genome as represented by the 840 short read accessions mapped to the graph. The fractions of the overlap have been calculated from both directions, and based on the number of nodes and the amount of sequence they represent in the graph. The core category of the mappings entail all nodes that are mapped to by at least 90% of the accessions, while the private category contains all nodes that are mapped to by 10% or fewer of the accessions.

	Nodes in the Graph				Sequence in the Graph [Mb]			
	core	*shell*	*private*	unmapped	*core*	*shell*	*private*	unmapped
core	1,365,373	1,149,616	304,244	2,776	57.4	10	2.2	0.3
shell	555,212	1,674,585	238,854	2,116	22.8	21.6	2.8	0.4
private	334,727	664,794	390,159	12,350	13.1	18.8	11.9	7.7
	As fraction of nodes covered in the mapping population							
core	60.5	33	32.6	0.161	61.5	19.8	12.9	3.8
shell	24.6	48	25.6	0.1227	24.4	42.9	16.6	4.9
private	14.8	19.1	41.8	0.7163	14.1	37.3	70.5	91.2
	As fraction node in the graph							
core	48.4	40.7	10.8	0.1	82.2	14.3	3.1	0.5
shell	22.5	67.8	9.7	0.1	47.9	45.4	5.9	0.9
private	23.9	47.4	27.8	0.9	25.5	36.5	23.1	14.9

Bibliography

1000 Genomes Project Consortium, Abecasis, G. R., Altshuler, D., Auton, A., Brooks, L. D., Durbin, R. M., Gibbs, R. A., Hurles, M. E., and McVean, G. A. (2010). A map of human genome variation from population-scale sequencing. *Nature*, **467**(7319), 1061–1073.

1001 Genomes Consortium (2016). 1,135 genomes reveal the global pattern of polymorphism in arabidopsis thaliana. *Cell*, **166**(2), 481–491.

Abel, H. J., Larson, D. E., Regier, A. A., Chiang, C., Das, I., Kanchi, K. L., Layer, R. M., Neale, B. M., Salerno, W. J., Reeves, C., Buyske, S., NHGRI Centers for Common Disease Genomics, Matise, T. C., Muzny, D. M., Zody, M. C., Lander, E. S., Dutcher, S. K., Stitziel, N. O., and Hall, I. M. (2020). Mapping and characterization of structural variation in 17,795 human genomes. *Nature*, **583**(7814), 83–89.

Adams, D. (1995). *The hitch hiker's guide to the galaxy: a trilogy in five parts*. Random House.

Adams, M. D., Celniker, S. E., Holt, R. A., Evans, C. A., Gocayne, J. D., Amanatides, P. G., Scherer, S. E., Li, P. W., Hoskins, R. A., Galle, R. F., George, R. A., Lewis, S. E., Richards, S., Ashburner, M., Henderson, S. N., Sutton, G. G., Wortman, J. R., Yandell, M. D., Zhang, Q., Chen, L. X., Brandon, R. C., Rogers, Y. H., Blazej, R. G., Champe, M., Pfeiffer, B. D., Wan, K. H., Doyle, C., Baxter, E. G., Helt, G., Nelson, C. R., Gabor, G. L., Abril, J. F., Agbayani, A., An, H. J., Andrews-Pfannkoch, C., Baldwin, D., Ballew, R. M., Basu, A., Baxendale, J., Bayraktaroglu, L., Beasley, E. M., Beeson, K. Y., Benos, P. V., Berman, B. P., Bhandari, D., Bolshakov, S., Borkova, D., Botchan, M. R., Bouck, J., Brokstein, P., Brottier, P., Burtis, K. C., Busam, D. A., Butler, H., Cadieu, E., Center, A., Chandra, I., Cherry, J. M., Cawley, S., Dahlke, C., Davenport, L. B., Davies, P., de Pablos, B., Delcher, A., Deng, Z., Mays, A. D., Dew, I., Dietz, S. M., Dodson, K., Doup, L. E., Downes, M., Dugan-Rocha, S., Dunkov, B. C., Dunn, P., Durbin, K. J., Evangelista, C. C., Ferraz, C., Ferriera, S., Fleischmann, W., Fosler, C., Gabrielian, A. E., Garg, N. S., Gelbart, W. M., Glasser, K., Glodek, A., Gong, F., Gorrell, J. H., Gu, Z., Guan, P., Harris, M., Harris, N. L., Harvey, D., Heiman, T. J., Hernandez, J. R., Houck, J., Hostin, D., Houston, K. A., Howland, T. J., Wei, M. H., Ibegwam, C., Jalali, M., Kalush, F., Karpen, G. H., Ke, Z., Kennison, J. A., Ketchum, K. A., Kimmel, B. E., Kodira, C. D., Kraft, C., Kravitz, S., Kulp, D., Lai, Z., Lasko, P., Lei, Y., Levitsky, A. A., Li, J., Li, Z., Liang, Y., Lin, X., Liu, X.,

Mattei, B., McIntosh, T. C., McLeod, M. P., McPherson, D., Merkulov, G., Milshina, N. V., Mobarry, C., Morris, J., Moshrefi, A., Mount, S. M., Moy, M., Murphy, B., Murphy, L., Muzny, D. M., Nelson, D. L., Nelson, D. R., Nelson, K. A., Nixon, K., Nusskern, D. R., Pacleb, J. M., Palazzolo, M., Pittman, G. S., Pan, S., Pollard, J., Puri, V., Reese, M. G., Reinert, K., Remington, K., Saunders, R. D., Scheeler, F., Shen, H., Shue, B. C., Sidén-Kiamos, I., Simpson, M., Skupski, M. P., Smith, T., Spier, E., Spradling, A. C., Stapleton, M., Strong, R., Sun, E., Svirskas, R., Tector, C., Turner, R., Venter, E., Wang, A. H., Wang, X., Wang, Z. Y., Wassarman, D. A., Weinstock, G. M., Weissenbach, J., Williams, S. M., WoodageT, Worley, K. C., Wu, D., Yang, S., Yao, Q. A., Ye, J., Yeh, R. F., Zaveri, J. S., Zhan, M., Zhang, G., Zhao, Q., Zheng, L., Zheng, X. H., Zhong, F. N., Zhong, W., Zhou, X., Zhu, S., Zhu, X., Smith, H. O., Gibbs, R. A., Myers, E. W., Rubin, G. M., and Venter, J. C. (2000). The genome sequence of drosophila melanogaster. *Science*, **287**(5461), 2185–2195.

Al-Mssallem, I. S., Hu, S., Zhang, X., Lin, Q., Liu, W., Tan, J., Yu, X., Liu, J., Pan, L., Zhang, T., Yin, Y., Xin, C., Wu, H., Zhang, G., Ba Abdullah, M. M., Huang, D., Fang, Y., Alnakhli, Y. O., Jia, S., Yin, A., Alhuzimi, E. M., Alsaihati, B. A., Al-Owayyed, S. A., Zhao, D., Zhang, S., Al-Otaibi, N. A., Sun, G., Majrashi, M. A., Li, F., Tala, Wang, J., Yun, Q., Alnassar, N. A., Wang, L., Yang, M., Al-Jelaify, R. F., Liu, K., Gao, S., Chen, K., Alkhaldi, S. R., Liu, G., Zhang, M., Guo, H., and Yu, J. (2013). Genome sequence of the date palm phoenix dactylifera L. *Nat. Commun.*, **4**, 2274.

Alonge, M., Wang, X., Benoit, M., Soyk, S., Pereira, L., Zhang, L., Suresh, H., Ramakrishnan, S., Maumus, F., Ciren, D., Levy, Y., Harel, T. H., Shalev-Schlosser, G., Amsellem, Z., Razifard, H., Caicedo, A. L., Tieman, D. M., Klee, H., Kirsche, M., Aganezov, S., Ranallo-Benavidez, T. R., Lemmon, Z. H., Kim, J., Robitaille, G., Kramer, M., Goodwin, S., McCombie, W. R., Hutton, S., Van Eck, J., Gillis, J., Eshed, Y., Sedlazeck, F. J., van der Knaap, E., Schatz, M. C., and Lippman, Z. B. (2020). Major impacts of widespread structural variation on gene expression and crop improvement in tomato. *Cell*, **182**(1), 145–161.e23.

Ameur, A. (2019). Goodbye reference, hello genome graphs. *Nat. Biotechnol.*, **37**(8), 866–868.

Amin, M. R., Yurovsky, A., Tian, Y., and Skiena, S. (2018). DeepAnnotator: Genome annotation with deep learning. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '18, pages 254–259, New York, NY, USA. Association for Computing Machinery.

anaconda (2020). Anaconda software distribution.

Arabidopsis Genome Initiative (2000). Analysis of the genome sequence of the flowering plant arabidopsis thaliana. *Nature*, **408**(6814), 796–815.

Armstrong, J., Hickey, G., Diekhans, M., Fiddes, I. T., Novak, A. M., Deran, A., Fang, Q., Xie, D., Feng, S., Stiller, J., Genereux, D., Johnson, J., Marinescu, V. D., Alföldi, J., Harris, R. S., Lindblad-Toh, K., Haussler, D., Karlsson, E., Jarvis, E. D., Zhang, G., and Paten, B. (2020). Progressive cactus is a multiple-genome aligner for the thousand-genome era. *Nature*, **587**(7833), 246–251.

Baduel, P., Leduque, B., Ignace, A., Gy, I., Gil, J., Loudet, O., Vincent, C., and Quadrana, L. (2021). Genetic and environmental modulation of transposition shapes the evolutionary potential of arabidopsis thaliana. `https://hal.archives-ouvertes.fr/hal-03099067/document`. Accessed: 2021-2-13.

Baer, R., Bankier, A. T., Biggin, M. D., Deininger, P. L., Farrell, P. J., Gibson, T. J., Hatfull, G., Hudson, G. S., Satchwell, S. C., and Séguin, C. (1984). DNA sequence and expression of the B95-8 Epstein-Barr virus genome. *Nature*, **310**(5974), 207–211.

Ballouz, S., Dobin, A., and Gillis, J. A. (2019). Is it time to change the reference genome? *Genome Biol.*, **20**(1), 159.

Bankevich, A., Bzikadze, A., Kolmogorov, M., Antipov, D., and Pevzner, P. A. (2021). LJA: Assembling long and accurate reads using multiplex de bruijn graphs.

Bayer, P. E., Petereit, J., Durant, É., Monat, C., Rouard, M., Hu, H., Chapman, B., Li, C., Cheng, S., Batley, J., and Edwards, D. (2022). Wheat panache - a pangenome graph database representing presence/absence variation across 16 bread wheat genomes.

Berardini, T. Z., Reiser, L., Li, D., Mezheritsky, Y., Muller, R., Strait, E., and Huala, E. (2015). The arabidopsis information resource: Making and mining the "gold standard" annotated reference plant genome. *Genesis*, **53**(8), 474–485.

Beyer, W., Novak, A. M., Hickey, G., Chan, J., Tan, V., Paten, B., and Zerbino, D. R. (2019). Sequence tube maps: making graph genomes intuitive to commuters. *Bioinformatics*, **35**(24), 5318–5320.

Beyter, D., Ingimundardottir, H., Oddsson, A., Eggertsson, H. P., Bjornsson, E., Jonsson, H., Atlason, B. A., Kristmundsdottir, S., Mehringer, S., Hardarson, M. T., Gudjonsson, S. A., Magnusdottir, D. N., Jonasdottir, A., Jonasdottir, A., Kristjansson, R. P., Sverrisson, S. T., Holley, G., Palsson, G., Stefansson, O. A., Eyjolfsson, G., Olafsson, I., Sigurdardottir, O., Torfason, B., Masson, G., Helgason, A., Thorsteinsdottir, U., Holm, H., Gudbjartsson, D. F., Sulem, P., Magnusson, O. T., Halldorsson, B. V., and Stefansson, K. (2020). Long read sequencing of 3,622 icelanders provides insight into the role of structural variants in human diseases and other traits.

Blattner, F. R., Plunkett, 3rd, G., Bloch, C. A., Perna, N. T., Burland, V., Riley, M., Collado-Vides, J., Glasner, J. D., Rode, C. K., Mayhew, G. F., Gregor, J., Davis, N. W.,

Kirkpatrick, H. A., Goeden, M. A., Rose, D. J., Mau, B., and Shao, Y. (1997). The complete genome sequence of escherichia coli K-12. *Science*, **277**(5331), 1453–1462.

Brázda, V., Bohálová, N., and Bowater, R. P. (2022). New telomere to telomere assembly of human chromosome 8 reveals a previous underestimation of g-quadruplex forming sequences and inverted repeats. *Gene*, **810**, 146058.

Breitwieser, F. P. and Salzberg, S. L. (2020). Pavian: interactive analysis of metagenomics data for microbiome studies and pathogen identification. *Bioinformatics*, **36**(4), 1303–1304.

Buchfink, B., Reuter, K., and Drost, H.-G. (2021). Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nat. Methods*, **18**(4), 366–368.

Budiman, M. A., Mao, L., Wood, T. C., and Wing, R. A. (2000). A deep-coverage tomato BAC library and prospects toward development of an STC framework for genome sequencing. *Genome Res.*, **10**(1), 129–136.

C. elegans Sequencing Consortium (1998). Genome sequence of the nematode c. elegans: a platform for investigating biology. *Science*, **282**(5396), 2012–2018.

Caicedo, A. L., Williamson, S. H., Hernandez, R. D., Boyko, A., Fledel-Alon, A., York, T. L., Polato, N. R., Olsen, K. M., Nielsen, R., McCouch, S. R., Bustamante, C. D., and Purugganan, M. D. (2007). Genome-wide patterns of nucleotide polymorphism in domesticated rice. *PLoS Genet.*, **3**(9), 1745–1756.

Cameron, D. L., Baber, J., Shale, C., Valle-Inclan, J. E., Besselink, N., van Hoeck, A., Janssen, R., Cuppen, E., Priestley, P., and Papenfuss, A. T. (2021). GRIDSS2: comprehensive characterisation of somatic structural variation using single breakend variants and structural variant phasing. *Genome Biol.*, **22**(1), 202.

Chatterjee, N. and Walker, G. C. (2017). Mechanisms of DNA damage, repair, and mutagenesis. *Environ. Mol. Mutagen.*, **58**(5), 235–263.

Chen, J., Wang, Z., Tan, K., Huang, W., Shi, J., Li, T., Hu, J., Wang, K., Wang, C., Xin, B., Zhao, H., Song, W., Hufford, M. B., Schnable, J. C., Jin, W., and Lai, J. (2023). A complete telomere-to-telomere assembly of the maize genome. *Nat. Genet.*, **55**(7), 1221–1231.

Chen, N.-C., Solomon, B., Mun, T., Iyer, S., and Langmead, B. (2021). Reference flow: reducing reference bias using multiple population genomes. *Genome Biol.*, **22**(1), 8.

Cheng, C.-Y., Krishnakumar, V., Chan, A. P., Thibaud-Nissen, F., Schobel, S., and Town, C. D. (2017). Araport11: a complete reannotation of the arabidopsis thaliana reference genome. *Plant J.*, **89**(4), 789–804.

Chin, C.-S., Alexander, D. H., Marks, P., Klammer, A. A., Drake, J., Heiner, C., Clum, A., Copeland, A., Huddleston, J., Eichler, E. E., Turner, S. W., and Korlach, J. (2013). Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nat. Methods*, **10**(6), 563–569.

Chin, C.-S., Behera, S., Khalak, A., Sedlazeck, F. J., Sudmant, P. H., Wagner, J., and Zook, J. M. (2023). Multiscale analysis of pangenomes enables improved representation of genomic diversity for repetitive and clinically relevant genes. *Nat. Methods*.

Choi, J. Y., Lye, Z. N., Groen, S. C., Dai, X., Rughani, P., Zaaijer, S., Harrington, E. D., Juul, S., and Purugganan, M. D. (2020). Nanopore sequencing-based genome assembly and evolutionary genomics of circum-basmati rice. *Genome Biol.*, **21**(1), 21.

Cristina Barragan, A., Collenberg, M., Schwab, R., Kerstens, M., Bezrukov, I., Bemm, F., Požárová, D., Kolář, F., and Weigel, D. (2021). Homozygosity at its limit: Inbreeding depression in wild arabidopsis arenosa populations.

Crysnanto, D. and Pausch, H. (2019). Sequence read mapping and variant discovery from bovine breed-specific augmented reference graphs.

Dabbaghie, F., Ebler, J., and Marschall, T. (2021). BubbleGun: Enumerating bubbles and superbubbles in genome graphs.

Danecek, P., Auton, A., Abecasis, G., Albers, C. A., Banks, E., DePristo, M. A., Handsaker, R. E., Lunter, G., Marth, G. T., Sherry, S. T., McVean, G., Durbin, R., and 1000 Genomes Project Analysis Group (2011). The variant call format and VCFtools. *Bioinformatics*, **27**(15), 2156–2158.

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., Whitwham, A., Keane, T., McCarthy, S. A., Davies, R. M., and Li, H. (2021). Twelve years of SAMtools and BCFtools. *Gigascience*, **10**(2).

De Coster, W. and Van Broeckhoven, C. (2019). Newest methods for detecting structural variations. *Trends Biotechnol.*, **37**(9), 973–982.

Degner, J. F., Marioni, J. C., Pai, A. A., Pickrell, J. K., Nkadori, E., Gilad, Y., and Pritchard, J. K. (2009). Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data. *Bioinformatics*, **25**(24), 3207–3212.

del Pozo, J. C. and Ramirez-Parra, E. (2015). Whole genome duplications in plants: an overview from arabidopsis. *J. Exp. Bot.*, **66**(22), 6991–7003.

Denti, L., Rizzi, R., Beretta, S., Vedova, G. D., Previtali, M., and Bonizzoni, P. (2018). ASGAL: aligning RNA-Seq data to a splicing graph to detect novel alternative splicing events. *BMC Bioinformatics*, **19**(1), 444.

Dewey, F. E., Chen, R., Cordero, S. P., Ormond, K. E., Caleshu, C., Karczewski, K. J., Whirl-Carrillo, M., Wheeler, M. T., Dudley, J. T., Byrnes, J. K., Cornejo, O. E., Knowles, J. W., Woon, M., Sangkuhl, K., Gong, L., Thorn, C. F., Hebert, J. M., Capriotti, E., David, S. P., Pavlovic, A., West, A., Thakuria, J. V., Ball, M. P., Zaranek, A. W., Rehm, H. L., Church, G. M., West, J. S., Bustamante, C. D., Snyder, M., Altman, R. B., Klein, T. E., Butte, A. J., and Ashley, E. A. (2011). Phased whole-genome genetic risk in a family quartet using a major allele reference sequence. *PLoS Genet.*, **7**(9), e1002280.

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T. R. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, **29**(1), 15–21.

Dujon, B. (1996). The yeast genome project: what did we learn? *Trends Genet.*, **12**(7), 263–270.

Durant, É., Sabot, F., Conte, M., and Rouard, M. (2021). Panache: a web Browser-Based viewer for linearized pangenomes.

Durvasula, A., Fulgione, A., Gutaker, R. M., Alacakaptan, S. I., Flood, P. J., Neto, C., Tsuchimatsu, T., Burbano, H. A., Picó, F. X., Alonso-Blanco, C., and Hancock, A. M. (2017). African genomes illuminate the early history and transition to selfing in arabidopsis thaliana. *Proc. Natl. Acad. Sci. U. S. A.*, **114**(20), 5213–5218.

Ebler, J., Clarke, W. E., Rausch, T., Audano, P. A., Houwaart, T., Korbel, J., Eichler, E. E., Zody, M. C., Dilthey, A. T., and Marschall, T. (2020). Pangenome-based genome inference.

Emms, D. M. and Kelly, S. (2015). OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. *Genome Biol.*, **16**, 157.

Emms, D. M. and Kelly, S. (2019). OrthoFinder: phylogenetic orthology inference for comparative genomics. *Genome Biol.*, **20**(1), 238.

Fan, X., Abbott, T. E., Larson, D., and Chen, K. (2014). BreakDancer: Identification of genomic structural variation from Paired-End read mapping. *Curr. Protoc. Bioinformatics*, **45**, 15.6.1–11.

Fiers, W., Contreras, R., Duerinck, F., Haegeman, G., Iserentant, D., Merregaert, J., Min Jou, W., Molemans, F., Raeymaekers, A., Van den Berghe, A., Volckaert, G., and Ysebaert, M. (1976). Complete nucleotide sequence of bacteriophage MS2 RNA: primary and secondary structure of the replicase gene. *Nature*, **260**(5551), 500–507.

Fransz, P., Linc, G., Lee, C.-R., Aflitos, S. A., Lasky, J. R., Toomajian, C., Ali, H., Peters, J., van Dam, P., Ji, X., Kuzak, M., Gerats, T., Schubert, I., Schneeberger, K., Colot, V., Martienssen, R., Koornneef, M., Nordborg, M., Juenger, T. E., de Jong, H., and Schranz, M. E. (2016). Molecular, genetic and evolutionary analysis of a paracentric inversion in arabidopsis thaliana. *Plant J.*, **88**(2), 159–178.

Fulgione, A., Koornneef, M., Roux, F., Hermisson, J., and Hancock, A. M. (2018). Madeiran arabidopsis thaliana reveals ancient Long-Range colonization and clarifies demography in eurasia. *Mol. Biol. Evol.*, **35**(3), 564–574.

Gao, L., Gonda, I., Sun, H., Ma, Q., Bao, K., Tieman, D. M., Burzynski-Chang, E. A., Fish, T. L., Stromberg, K. A., Sacks, G. L., Thannhauser, T. W., Foolad, M. R., Diez, M. J., Blanca, J., Canizares, J., Xu, Y., van der Knaap, E., Huang, S., Klee, H. J., Giovannoni, J. J., and Fei, Z. (2019). The tomato pan-genome uncovers new genes and a rare allele regulating fruit flavor. *Nat. Genet.*, **51**(6), 1044–1051.

Garrison, E. and Guarracino, A. (2023). Unbiased pangenome graphs. *Bioinformatics*, **39**(1).

Garrison, E. and Marth, G. (2012). Haplotype-based variant detection from short-read sequencing. *arXiv*.

Garrison, E., Sirén, J., Novak, A. M., Hickey, G., Eizenga, J. M., Dawson, E. T., Jones, W., Garg, S., Markello, C., Lin, M. F., Paten, B., and Durbin, R. (2018). Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nat. Biotechnol.*, **36**(9), 875–879.

Garrison, E., Kronenberg, Z. N., Dawson, E. T., Pedersen, B. S., and Prins, P. (2021). Vcflib and tools for processing the VCF variant call format.

Garrison, E., Guarracino, A., Heumos, S., Villani, F., Bao, Z., Tattini, L., Hagmann, J., Vorbrugg, S., Marco-Sola, S., Kubica, C., Ashbrook, D. G., Thorell, K., Rusholme-Pilcher, R. L., Liti, G., Rudbeck, E., Nahnsen, S., Yang, Z., Moses, M. N., Nobrega, F. L., Wu, Y., Chen, H., de Ligt, J., Sudmant, P. H., Soranzo, N., Colonna, V., Williams, R. W., and Prins, P. (2023). Building pangenome graphs.

GFA (2022). Gfa: Graphical fragment assembly (gfa) format specification.

Giguere, D. J., Bahcheli, A. T., Slattery, S. S., Patel, R. R., Flatley, M., Karas, B. J., Edgell, D. R., and Gloor, G. B. (2021). Telomere-to-telomere genome assembly of phaeodactylum tricornutum.

Goel, M. and Schneeberger, K. (2022). plotsr: Visualising structural similarities and rearrangements between multiple genomes.

Goel, M., Sun, H., Jiao, W.-B., and Schneeberger, K. (2019). SyRI: finding genomic rearrangements and local sequence differences from whole-genome assemblies. *Genome Biol.*, **20**(1), 277.

Goffeau, A., Barrell, B. G., Bussey, H., Davis, R. W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J. D., Jacq, C., Johnston, M., Louis, E. J., Mewes, H. W., Murakami, Y., Philippsen, P., Tettelin, H., and Oliver, S. G. (1996). Life with 6000 genes. *Science*, **274**(5287), 546, 563–7.

Göktay, M., Fulgione, A., and Hancock, A. M. (2020). A new catalogue of structural variants in 1301 a. thaliana lines from africa, eurasia and north america reveals a signature of balancing at defense response genes. *Mol. Biol. Evol.*

Gonnella, G., Niehus, N., and Kurtz, S. (2019). GfaViz: flexible and interactive visualization of GFA sequence graphs. *Bioinformatics*, **35**(16), 2853–2855.

Gonzalez de la Rosa, P. M., Thomson, M., Trivedi, U., Tracey, A., Tandonnet, S., and Blaxter, M. (2021). A telomere-to-telomere assembly of oscheius tipulae and the evolution of rhabditid nematode chromosomes. *G3*, **11**(1).

Groza, C., Chen, X., Pacis, A., Simon, M.-M., Pramatarova, A., Aracena, K. A., Pastinen, T., Barreiro, L. B., and Bourque, G. (2021). Genome graphs detect human polymorphisms in active epigenomic states during influenza infection.

Grytten, I., Rand, K. D., Nederbragt, A. J., and Sandve, G. K. (2020). Assessing graph-based read mappers against a baseline approach highlights strengths and weaknesses of current methods. *BMC Genomics*, **21**(1), 282.

Grytten, I., Rand, K. D., and Sandve, G. K. (2021). KAGE: Fast alignment-free graph-based genotyping of SNPs and short indels.

Guo, S., Zhao, S., Sun, H., Wang, X., Wu, S., Lin, T., Ren, Y., Gao, L., Deng, Y., Zhang, J., Lu, X., Zhang, H., Shang, J., Gong, G., Wen, C., He, N., Tian, S., Li, M., Liu, J., Wang, Y., Zhu, Y., Jarret, R., Levi, A., Zhang, X., Huang, S., Fei, Z., Liu, W., and Xu, Y. (2019). Resequencing of 414 cultivated and wild watermelon accessions identifies selection for fruit quality traits. *Nat. Genet.*, **51**(11), 1616–1623.

Gupta, P. K. (2021). GWAS for genetics of complex quantitative traits: Genome to pangenome and SNPs to SVs and k-mers. *Bioessays*, page e2100109.

Haas, B. J., Salzberg, S. L., Zhu, W., Pertea, M., Allen, J. E., Orvis, J., White, O., Buell, C. R., and Wortman, J. R. (2008). Automated eukaryotic gene structure annotation using EVidenceModeler and the program to assemble spliced alignments. *Genome Biol.*, **9**(1), R7.

Hadi, K., Yao, X., Behr, J. M., Deshpande, A., Xanthopoulakis, C., Tian, H., Kudman, S., Rosiene, J., Darmofal, M., DeRose, J., Mortensen, R., Adney, E. M., Shaiber, A., Gajic, Z., Sigouros, M., Eng, K., Wala, J. A., Wrzeszczyński, K. O., Arora, K., Shah, M., Emde, A.-K., Felice, V., Frank, M. O., Darnell, R. B., Ghandi, M., Huang, F., Dewhurst, S., Maciejowski, J., de Lange, T., Setton, J., Riaz, N., Reis-Filho, J. S., Powell, S., Knowles, D. A., Reznik, E., Mishra, B., Beroukhim, R., Zody, M. C., Robine, N., Oman, K. M., Sanchez, C. A., Kuhner, M. K., Smith, L. P., Galipeau, P. C., Paulson, T. G., Reid, B. J., Li, X., Wilkes, D., Sboner, A., Mosquera, J. M., Elemento, O., and Imielinski, M. (2020). Distinct classes of complex structural variation uncovered across thousands of cancer genome graphs. *Cell*, **183**(1), 197–210.e32.

He, Q., Tang, S., Zhi, H., Chen, J., Zhang, J., Liang, H., Alam, O., Li, H., Zhang, H., Xing, L., Li, X., Zhang, W., Wang, H., Shi, J., Du, H., Wu, H., Wang, L., Yang, P., Xing, L., Yan, H., Song, Z., Liu, J., Wang, H., Tian, X., Qiao, Z., Feng, G., Guo, R., Zhu, W., Ren, Y., Hao, H., Li, M., Zhang, A., Guo, E., Yan, F., Li, Q., Liu, Y., Tian, B., Zhao, X., Jia, R., Feng, B., Zhang, J., Wei, J., Lai, J., Jia, G., Purugganan, M., and Diao, X. (2023). A graph-based genome and pan-genome variation of the model plant setaria. *Nat. Genet.*, **55**(7), 1232–1242.

Hickey, G., Heller, D., Monlong, J., Sibbesen, J. A., Sirén, J., Eizenga, J., Dawson, E. T., Garrison, E., Novak, A. M., and Paten, B. (2020). Genotyping structural variants in pangenome graphs using the vg toolkit. *Genome Biol.*, **21**(1), 35.

Hirsch, C. N., Foerster, J. M., Johnson, J. M., Sekhon, R. S., Muttoni, G., Vaillancourt, B., Peñagaricano, F., Lindquist, E., Pedraza, M. A., Barry, K., de Leon, N., Kaeppler, S. M., and Buell, C. R. (2014). Insights into the maize pan-genome and pan-transcriptome. *Plant Cell*, **26**(1), 121–135.

Hirsch, C. N., Hirsch, C. D., Brohammer, A. B., Bowman, M. J., Soifer, I., Barad, O., Shem-Tov, D., Baruch, K., Lu, F., Hernandez, A. G., Fields, C. J., Wright, C. L., Koehler, K., Springer, N. M., Buckler, E., Buell, C. R., de Leon, N., Kaeppler, S. M., Childs, K. L., and Mikel, M. A. (2016). Draft assembly of elite inbred line PH207 provides insights into genomic and transcriptome diversity in maize. *Plant Cell*, **28**(11), 2700–2714.

Ho, S. S., Urban, A. E., and Mills, R. E. (2020). Structural variation in the sequencing era. *Nat. Rev. Genet.*, **21**(3), 171–189.

Hoff, K. J., Lomsadze, A., Borodovsky, M., and Stanke, M. (2019). Whole-Genome annotation with BRAKER. *Methods Mol. Biol.*, **1962**, 65–95.

Hollister, J. D., Smith, L. M., Guo, Y.-L., Ott, F., Weigel, D., and Gaut, B. S. (2011). Transposable elements and small RNAs contribute to gene expression divergence between arabidopsis thaliana and arabidopsis lyrata. *Proc. Natl. Acad. Sci. U. S. A.*, **108**(6), 2322–2327.

Hufford, M. B., Seetharam, A. S., Woodhouse, M. R., Chougule, K. M., Ou, S., Liu, J., Ricci, W. A., Guo, T., Olson, A., Qiu, Y., Della Coletta, R., Tittes, S., Hudson, A. I., Marand, A. P., Wei, S., Lu, Z., Wang, B., Tello-Ruiz, M. K., Piri, R. D., Wang, N., Kim, D. W., Zeng, Y., O'Connor, C. H., Li, X., Gilbert, A. M., Baggs, E., Krasileva, K. V., Portwood, J. L., Cannon, E. K. S., Andorf, C. M., Manchanda, N., Snodgrass, S. J., Hufnagel, D. E., Jiang, Q., Pedersen, S., Syring, M. L., Kudrna, D. A., Llaca, V., Fengler, K., Schmitz, R. J., Ross-Ibarra, J., Yu, J., Gent, J. I., Hirsch, C. N., Ware, D., and Kelly Dawe, R. (2021). De novo assembly, annotation, and comparative analysis of 26 diverse maize genomes.

Jackson, B. G., Regennitter, M., Yang, X., Schnable, P. S., and Aluru, S. (2010). Parallel de novo assembly of large genomes from high-throughput short reads. In *2010 IEEE International Symposium on Parallel Distributed Processing (IPDPS)*, pages 1–10.

Jaegle, B., Soto-Jiménez, L. M., Burns, R., Rabanal, F. A., and Nordborg, M. (2021). Extensive gene duplication in arabidopsis revealed by pseudo-heterozygosity.

Jayakodi, M., Padmarasu, S., Haberer, G., Bonthala, V. S., Gundlach, H., Monat, C., Lux, T., Kamal, N., Lang, D., Himmelbach, A., Ens, J., Zhang, X.-Q., Angessa, T. T., Zhou, G., Tan, C., Hill, C., Wang, P., Schreiber, M., Boston, L. B., Plott, C., Jenkins, J., Guo, Y., Fiebig, A., Budak, H., Xu, D., Zhang, J., Wang, C., Grimwood, J., Schmutz, J., Guo, G., Zhang, G., Mochida, K., Hirayama, T., Sato, K., Chalmers, K. J., Langridge, P., Waugh, R., Pozniak, C. J., Scholz, U., Mayer, K. F. X., Spannagl, M., Li, C., Mascher, M., and Stein, N. (2020). The barley pan-genome reveals the hidden legacy of mutation breeding. *Nature*, **588**(7837), 284–289.

Jiao, W.-B. and Schneeberger, K. (2020). Chromosome-level assemblies of multiple arabidopsis genomes reveal hotspots of rearrangements with altered evolutionary dynamics. *Nat. Commun.*, **11**(1), 989.

Jin, M., Liu, H., He, C., Fu, J., Xiao, Y., Wang, Y., Xie, W., Wang, G., and Yan, J. (2016). Maize pan-transcriptome provides novel insights into genome complexity and quantitative trait variation. *Sci. Rep.*, **6**, 18936.

Jordan, K. W., Bradbury, P. J., Miller, Z. R., Nyine, M., He, F., Fraser, M., Anderson, J., Mason, E., Katz, A., Pearce, S., Carter, A. H., Prather, S., Pumphrey, M., Chen, J., Cook, J., Liu, S., Rudd, J. C., Wang, Z., Chu, C., Ibrahim, A. M. H., Turkus, J., Olson, E., Nagarajan, R., Carver, B., Yan, L., Taagen, E., Sorrells, M., Ward, B., Ren, J., Akhunova, A., Bai, G., Bowden, R., Fiedler, J., Faris, J., Dubcovsky, J., Guttieri, M., Brown-Guedira, G., Buckler, E., Jannink, J.-L., and Akhunov, E. D. (2021). Development of the wheat practical haplotype graph database as a resource for genotyping data storage and genotype imputation.

Kawakatsu, T., Huang, S.-S. C., Jupe, F., Sasaki, E., Schmitz, R. J., Urich, M. A., Castanon, R., Nery, J. R., Barragan, C., He, Y., Chen, H., Dubin, M., Lee, C.-R., Wang, C., Bemm, F., Becker, C., O'Neil, R., O'Malley, R. C., Quarless, D. X., 1001 Genomes Consortium, Schork, N. J., Weigel, D., Nordborg, M., and Ecker, J. R. (2016). Epigenomic diversity in a global collection of arabidopsis thaliana accessions. *Cell*, **166**(2), 492–505.

Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., and Haussler, D. (2002). The human genome browser at UCSC. *Genome Res.*, **12**(6), 996–1006.

Kim, B. Y., Wang, J. R., Miller, D. E., Barmina, O., Delaney, E., Thompson, A., Comeault, A. A., Peede, D., D'Agostino, E. R. R., Pelaez, J., Aguilar, J. M., Haji, D., Matsunaga, T., Armstrong, E. E., Zych, M., Ogawa, Y., Stamenković-Radak, M., Jelić, M., Veselinović, M. S., Tanasković, M., Erić, P., Gao, J.-J., Katoh, T. K., Toda, M. J., Watabe, H., Watada, M., Davis, J. S., Moyle, L. C., Manoli, G., Bertolini, E., Košťál, V., Scott Hawley, R., Takahashi, A., Jones, C. D., Price, D. K., Whiteman, N., Kopp, A., Matute, D. R., and Petrov, D. A. (2020). Highly contiguous assemblies of 101 drosophilid genomes.

Kim, D., Paggi, J. M., Park, C., Bennett, C., and Salzberg, S. L. (2019). Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nat. Biotechnol.*, **37**(8), 907–915.

Koren, S., Walenz, B. P., Berlin, K., Miller, J. R., Bergman, N. H., and Phillippy, A. M. (2017). Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res.*, **27**(5), 722–736.

Korf, I. (2004). Gene finding in novel genomes. *BMC Bioinformatics*, **5**, 59.

Köster, J. and Rahmann, S. (2012). Building and documenting workflows with python-based snakemake. In *German Conference on Bioinformatics 2012*.

Kosugi, S., Momozawa, Y., Liu, X., Terao, C., Kubo, M., and Kamatani, Y. (2019). Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. *Genome Biol.*, **20**(1), 117.

Kris A. Wetterstrand, M. S. (2019). DNA sequencing costs: Data. `https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data`. Accessed: 2023-9-1.

Kubica, C. (2021). gfautils.

LeGault, L. H. and Dewey, C. N. (2013). Inference of alternative splicing from RNA-Seq data with probabilistic splice graphs. *Bioinformatics*, **29**(18), 2300–2310.

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv*.

Li, H. (2016). Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, **32**(14), 2103–2110.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, **34**(18), 3094–3100.

Li, H., Feng, X., and Chu, C. (2020). The design and construction of reference pangenome graphs with minigraph. *Genome Biol.*, **21**(1), 265.

Li, R., Gong, M., Zhang, X., Wang, F., Liu, Z., Zhang, L., Xu, M., Zhang, Y., Dai, X., Zhang, Z., Fang, W., Yang, Y., Zhang, H., Fu, W., Cao, C., Yang, P., Ghanatsaman, Z. A., Negari, N. J., Nanaei, H. A., Yue, X., Song, Y., Lan, X., Deng, W., Wang, X., Xiang, R., Ibeagha-Awemu, E. M.,) Heslop-Harrison, P. J., Lenstra, J. A., Gan, S., and Jiang, Y. (2022). The first sheep graph-based pan-genome reveals the spectrum of structural variations and their effects on tail phenotypes.

Linthorst, J., Hulsman, M., Holstege, H., and Reinders, M. (2015). Scalable multi whole-genome alignment using recursive exact matching.

Long, E. M., Bradbury, P. J., Cinta Romay, M., Buckler, E. S., and Robbins, K. R. (2021). Genome-wide imputation using the practical haplotype graph in the heterozygous crop cassava.

Lowry, D. B. and Willis, J. H. (2010). A widespread chromosomal inversion polymorphism contributes to a major life-history transition, local adaptation, and reproductive isolation. *PLoS Biol.*, **8**(9).

Lu, F., Romay, M. C., Glaubitz, J. C., Bradbury, P. J., Elshire, R. J., Wang, T., Li, Y., Li, Y., Semagn, K., Zhang, X., Hernandez, A. G., Mikel, M. A., Soifer, I., Barad, O., and Buckler, E. S. (2015). High-resolution genetic mapping of maize pan-genome sequence anchors. *Nat. Commun.*, **6**, 6914.

Ma, J., Cáceres, M., Salmela, L., Mäkinen, V., and Tomescu, A. I. (2022). GraphChainer: Co-linear chaining for accurate alignment of long reads to variation graphs.

Ma, X., Fan, J., Wu, Y., Zhao, S., Zheng, X., Sun, C., and Tan, L. (2020). Whole-genome de novo assemblies reveal extensive structural variations and dynamic organelle-to-nucleus DNA transfers in african and asian rice. *Plant J.*, **104**(3), 596–612.

Marco-Sola, S., Moure, J. C., Moreto, M., and Espinosa, A. (2021). Fast gap-affine pairwise alignment using the wavefront algorithm. *Bioinformatics*, **37**(4), 456–463.

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, **17**(1), 10–12.

Martiniano, R., Garrison, E., Jones, E. R., Manica, A., and Durbin, R. (2020). Removing reference bias and improving indel calling in ancient DNA data analysis by mapping to a sequence variation graph. *Genome Biol.*, **21**(1), 250.

Min Jou, W., Haegeman, G., Ysebaert, M., and Fiers, W. (1972). Nucleotide sequence of the gene coding for the bacteriophage MS2 coat protein. *Nature*, **237**(5350), 82–88.

Minkin, I. and Medvedev, P. (2020). Scalable multiple whole-genome alignment and locally collinear block construction with SibeliaZ. *Nat. Commun.*, **11**(1), 6327.

Mitchell (Mitch) McGrath, J., Funk, A., Galewski, P., Ou, S., Townsend, B., Davenport, K., Daligault, H., Johnson, S., Lee, J., Hastie, A., Darracq, A., Willems, G., Barnes, S., Liachko, I., Sullivan, S., Koren, S., Phillippy, A., Wang, J., Lu, T., Pulman, J., Childs, K., Yocum, A., Fermin, D., Mutasa-Göttgens, E., Stevanato, P., Taguchi, K., and Dorn, K. (2020). A contiguous de novo genome assembly of sugar beet EL10 (beta vulgaris l.).

Morrison, D. A. (2018). Multiple sequence alignment is not a solved problem. *arXiv*.

Mukamel, R. E., Handsaker, R. E., Sherman, M. A., Barton, A. R., Zheng, Y., McCarroll, S. A., and Loh, P.-R. (2021). Protein-coding repeat polymorphisms strongly shape diverse human phenotypes. *bioRxiv*, page 2021.01.19.427332.

Myers, E. W. (2005). The fragment assembly string graph. *Bioinformatics*, **21 Suppl 2**, ii79–85.

Nattestad, M. and Schatz, M. C. (2016). Assemblytics: a web analytics tool for the detection of variants from an assembly. *Bioinformatics*, **32**(19), 3021–3023.

Nguyen, N., Hickey, G., Zerbino, D. R., Raney, B., Earl, D., Armstrong, J., Kent, W. J., Haussler, D., and Paten, B. (2015). Building a pan-genome reference for a population. *J. Comput. Biol.*, **22**(5), 387–401.

Noshay, J. M., Marand, A. P., Anderson, S. N., Zhou, P., Guerra, M. K. M., Lu, Z., O'Connor, C., Crisp, P. A., Hirsch, C. N., Schmitz, R. J., and Springer, N. M. (2020). Cis-regulatory elements within TEs can influence expression of nearby maize genes.

Novak, A. M., Hickey, G., Garrison, E., Blum, S., Connelly, A., Dilthey, A., Eizenga, J., Saleh Elmohamed, M. A., Guthrie, S., Kahles, A., Keenan, S., Kelleher, J., Kural, D., Li, H., Lin, M. F., Miga, K., Ouyang, N., Rakocevic, G., Smuga-Otto, M., Zaranek, A. W., Durbin, R., McVean, G., Haussler, D., and Paten, B. (2017). Genome graphs.

Nowoshilow, S., Schloissnig, S., Fei, J.-F., Dahl, A., Pang, A. W. C., Pippel, M., Winkler, S., Hastie, A. R., Young, G., Roscito, J. G., Falcon, F., Knapp, D., Powell, S., Cruz, A., Cao, H., Habermann, B., Hiller, M., Tanaka, E. M., and Myers, E. W. (2018). The axolotl genome and the evolution of key tissue formation regulators. *Nature*, **554**(7690), 50–55.

O'Neil, R. (2016). structome: Toolbox development for structural variations within irys data retrieved from the BioNano genomics nick-labeling protocols.

Onodera, T., Sadakane, K., and Shibuya, T. (2013). Detecting superbubbles in assembly graphs. In *Algorithms in Bioinformatics*, pages 338–348. Springer Berlin Heidelberg.

Ou, S., Su, W., Liao, Y., Chougule, K., Agda, J. R. A., Hellinga, A. J., Lugo, C. S. B., Elliott, T. A., Ware, D., Peterson, T., Jiang, N., Hirsch, C. N., and Hufford, M. B. (2019). Benchmarking transposable element annotation methods for creation of a streamlined, comprehensive pipeline. *Genome Biol.*, **20**(1), 275.

pangenome consortium (2023). smoothxg: linearize and simplify variation graphs using blocked partial order alignment.

Paten, B., Earl, D., Nguyen, N., Diekhans, M., Zerbino, D., and Haussler, D. (2011). Cactus: Algorithms for genome multiple sequence alignment. *Genome Res.*, **21**(9), 1512–1528.

Paten, B., Eizenga, J. M., Rosen, Y. M., Novak, A. M., Garrison, E., and Hickey, G. (2018). Superbubbles, ultrabubbles, and cacti. *J. Comput. Biol.*, **25**(7), 649–663.

Pertea, G. and Pertea, M. (2020). GFF utilities: GffRead and GffCompare. *F1000Res.*, **9**.

Pevzner, P. A., Tang, H., and Waterman, M. S. (2001). An eulerian path approach to DNA fragment assembly. *Proc. Natl. Acad. Sci. U. S. A.*, **98**(17), 9748–9753.

Poplin, R., Ruano-Rubio, V., DePristo, M. A., Fennell, T. J., Carneiro, M. O., Van der Auwera, G. A., Kling, D. E., Gauthier, L. D., Levy-Moonshine, A., Roazen, D., Shakir, K., Thibault, J., Chandran, S., Whelan, C., Lek, M., Gabriel, S., Daly, M. J., Neale, B., MacArthur, D. G., and Banks, E. (2018). Scaling accurate genetic variant discovery to tens of thousands of samples.

Pritt, J., Chen, N.-C., and Langmead, B. (2018). FORGe: prioritizing variants for graph genomes. *Genome Biol.*, **19**(1), 220.

Quinlan, A. R. and Hall, I. M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**(6), 841–842.

Rausch, T., Zichner, T., Schlattl, A., Stütz, A. M., Benes, V., and Korbel, J. O. (2012). DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, **28**(18), i333–i339.

Rautiainen, M. and Marschall, T. (2019). GraphAligner: Rapid and versatile Sequence-to-Graph alignment.

Rogers, M. F., Thomas, J., Reddy, A. S., and Ben-Hur, A. (2012). SpliceGrapher: detecting patterns of alternative splicing from RNA-Seq data in the context of gene models and EST data. *Genome Biol.*, **13**(1), R4.

Rowan, B. A., Heavens, D., Feuerborn, T. R., Tock, A. J., Henderson, I. R., and Weigel, D. (2019). An ultra High-Density arabidopsis thaliana crossover map that refines the influences of structural variation and epigenetic features. *Genetics*, **213**(3), 771–787.

Sanger, F. and Thompson, E. O. P. (1953). The amino-acid sequence in the glycyl chain of insulin. II. the investigation of peptides from enzymic hydrolysates. *Biochem. J*, **53**(3), 366–374.

Schneeberger, K., Hagmann, J., Ossowski, S., Warthmann, N., Gesing, S., Kohlbacher, O., and Weigel, D. (2009). Simultaneous alignment of short reads against multiple genomes. *Genome Biol.*, **10**(9), R98.

Schneider, V. A., Graves-Lindsay, T., Howe, K., Bouk, N., Chen, H.-C., Kitts, P. A., Murphy, T. D., Pruitt, K. D., Thibaud-Nissen, F., Albracht, D., Fulton, R. S., Kremitzki, M., Magrini, V., Markovic, C., McGrath, S., Steinberg, K. M., Auger, K., Chow, W., Collins, J., Harden, G., Hubbard, T., Pelan, S., Simpson, J. T., Threadgold, G., Torrance, J., Wood, J., Clarke, L., Koren, S., Boitano, M., Li, H., Chin, C.-S., Phillippy, A. M., Durbin, R., Wilson, R. K., Flicek, P., and Church, D. M. (2016). Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly.

Seppey, M., Manni, M., and Zdobnov, E. M. (2019). BUSCO: Assessing genome assembly and annotation completeness. In M. Kollmar, editor, *Gene Prediction: Methods and Protocols*, pages 227–245. Springer New York, New York, NY.

Serhat Tetikol, H., Narci, K., Turgut, D., Budak, G., Kalay, O., Arslan, E., Demirkaya-Budak, S., Dolgoborodov, A., Jain, A., Kabakci-Zorlu, D., Brown, R., Semenyuk, V., and Davis-Dusenbery, B. (2021). Population-specific genome graphs improve high-throughput sequencing data analysis: A case study on the Pan-African genome.

Shi, T., Zhang, X., Hou, Y., Jiang, Y., Jia, C., Lai, Q., Dan, X., Feng, J., Feng, J., Ma, T., Wu, J., Liu, S., Zhang, L., Long, Z., Zhang, Y., Zhang, J., Chen, L., Street, N. R., Ingvarsson, P. K., Liu, J., Yin, T., and Wang, J. (2023). The super-pangenome of

populus unveil genomic facets for adaptation and diversification in widespread forest trees.

Shukla, H. G., Bawa, P. S., and Srinivasan, S. (2019). hg19KIndel: ethnicity normalized human reference genome. *BMC Genomics*, **20**(1), 459.

Shumate, A. and Salzberg, S. L. (2020). Liftoff: an accurate gene annotation mapping tool.

Simillion, C., Vandepoele, K., Van Montagu, M. C. E., Zabeau, M., and Van de Peer, Y. (2002). The hidden duplication past of arabidopsis thaliana. *Proc. Natl. Acad. Sci. U. S. A.*, **99**(21), 13627–13632.

Sirén, J. (2016). Indexing variation graphs. *arXiv*.

Sirén, J., Monlong, J., Chang, X., Novak, A. M., Eizenga, J. M., Markello, C., Sibbesen, J., Hickey, G., Chang, P.-C., Carroll, A., Haussler, D., Garrison, E., and Paten, B. (2020). Genotyping common, large structural variations in 5,202 genomes using pangenomes, the giraffe mapper, and the vg toolkit.

Slater, G. S. C. and Birney, E. (2005). Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*, **6**, 31.

Smit, A. F. A. and Hubley, R. (2008). RepeatModeler open-1.0.

Smit, AFA, Hubley, R & Green, P. (2013). RepeatMasker open-4.0. `http://www.repeatmasker.org`.

Sodeik, B., Doms, R. W., Ericsson, M., Hiller, G., Machamer, C. E., van Meer, G., Moss, B., and Griffiths, G. (1993). Assembly of vaccinia virus: role of the intermediate compartment between the endoplasmic reticulum and the golgi stacks. *J. Cell Biol.*, **121**(3), 521–541.

Solares, E. A., Chakraborty, M., Miller, D. E., Kalsow, S., Hall, K., Perera, A. G., Emerson, J. J., and Hawley, R. S. (2018). Rapid Low-Cost assembly of the drosophila melanogaster reference genome using Low-Coverage, Long-Read sequencing. *G3*, **8**(10), 3143–3154.

Song, B., Marco-Sola, S., Moreto, M., Johnson, L., Buckler, E. S., and Stitzer, M. C. (2022). AnchorWave: Sensitive alignment of genomes with high sequence diversity, extensive structural polymorphism, and whole-genome duplication. *Proc. Natl. Acad. Sci. U. S. A.*, **119**(1).

Soyk, S., Lemmon, Z. H., Sedlazeck, F. J., Jiménez-Gómez, J. M., Alonge, M., Hutton, S. F., Van Eck, J., Schatz, M. C., and Lippman, Z. B. (2019). Duplication of a domestication locus neutralized a cryptic variant that caused a breeding barrier in tomato. *Nat Plants*, **5**(5), 471–479.

Stanke, M., Diekhans, M., Baertsch, R., and Haussler, D. (2008). Using native and syntenically mapped cDNA alignments to improve de novo gene finding. *Bioinformatics*, **24**(5), 637–644.

Stevens, K. A., Wegrzyn, J. L., Zimin, A., Puiu, D., Crepeau, M., Cardeno, C., Paul, R., Gonzalez-Ibeas, D., Koriabine, M., Holtz-Morris, A. E., Martínez-García, P. J., Sezen, U. U., Marçais, G., Jermstad, K., McGuire, P. E., Loopstra, C. A., Davis, J. M., Eckert, A., de Jong, P., Yorke, J. A., Salzberg, S. L., Neale, D. B., and Langley, C. H. (2016). Sequence of the sugar pine megagenome. *Genetics*, **204**(4), 1613–1626.

Stoler, N. and Nekrutenko, A. (2021). Sequencing error profiles of illumina sequencing instruments. *NAR Genom Bioinform*, **3**(1), lqab019.

Studer, A., Zhao, Q., Ross-Ibarra, J., and Doebley, J. (2011). Identification of a functional transposon insertion in the maize domestication gene tb1. *Nat. Genet.*, **43**(11), 1160–1163.

Sudmant, P. H., Rausch, T., Gardner, E. J., Handsaker, R. E., Abyzov, A., Huddleston, J., Zhang, Y., Ye, K., Jun, G., Fritz, M. H.-Y., Konkel, M. K., Malhotra, A., Stütz, A. M., Shi, X., Casale, F. P., Chen, J., Hormozdiari, F., Dayama, G., Chen, K., Malig, M., Chaisson, M. J. P., Walter, K., Meiers, S., Kashin, S., Garrison, E., Auton, A., Lam, H. Y. K., Mu, X. J., Alkan, C., Antaki, D., Bae, T., Cerveira, E., Chines, P., Chong, Z., Clarke, L., Dal, E., Ding, L., Emery, S., Fan, X., Gujral, M., Kahveci, F., Kidd, J. M., Kong, Y., Lameijer, E.-W., McCarthy, S., Flicek, P., Gibbs, R. A., Marth, G., Mason, C. E., Menelaou, A., Muzny, D. M., Nelson, B. J., Noor, A., Parrish, N. F., Pendleton, M., Quitadamo, A., Raeder, B., Schadt, E. E., Romanovitch, M., Schlattl, A., Sebra, R., Shabalin, A. A., Untergasser, A., Walker, J. A., Wang, M., Yu, F., Zhang, C., Zhang, J., Zheng-Bradley, X., Zhou, W., Zichner, T., Sebat, J., Batzer, M. A., McCarroll, S. A., 1000 Genomes Project Consortium, Mills, R. E., Gerstein, M. B., Bashir, A., Stegle, O., Devine, S. E., Lee, C., Eichler, E. E., and Korbel, J. O. (2015). An integrated map of structural variation in 2,504 human genomes. *Nature*, **526**(7571), 75–81.

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J., and Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.*, **28**(5), 511–515.

Van de Weyer, A.-L., Monteiro, F., Furzer, O. J., Nishimura, M. T., Cevik, V., Witek, K., Jones, J. D. G., Dangl, J. L., Weigel, D., and Bemm, F. (2019). A Species-Wide inventory of NLR genes and alleles in arabidopsis thaliana. *Cell*, **178**(5), 1260–1272.e14.

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A., Gocayne, J. D., Amanatides, P., Ballew,

R. M., Huson, D. H., Wortman, J. R., Zhang, Q., Kodira, C. D., Zheng, X. H., Chen, L., Skupski, M., Subramanian, G., Thomas, P. D., Zhang, J., Gabor Miklos, G. L., Nelson, C., Broder, S., Clark, A. G., Nadeau, J., McKusick, V. A., Zinder, N., Levine, A. J., Roberts, R. J., Simon, M., Slayman, C., Hunkapiller, M., Bolanos, R., Delcher, A., Dew, I., Fasulo, D., Flanigan, M., Florea, L., Halpern, A., Hannenhalli, S., Kravitz, S., Levy, S., Mobarry, C., Reinert, K., Remington, K., Abu-Threideh, J., Beasley, E., Biddick, K., Bonazzi, V., Brandon, R., Cargill, M., Chandramouliswaran, I., Charlab, R., Chaturvedi, K., Deng, Z., Di Francesco, V., Dunn, P., Eilbeck, K., Evangelista, C., Gabrielian, A. E., Gan, W., Ge, W., Gong, F., Gu, Z., Guan, P., Heiman, T. J., Higgins, M. E., Ji, R. R., Ke, Z., Ketchum, K. A., Lai, Z., Lei, Y., Li, Z., Li, J., Liang, Y., Lin, X., Lu, F., Merkulov, G. V., Milshina, N., Moore, H. M., Naik, A. K., Narayan, V. A., Neelam, B., Nusskern, D., Rusch, D. B., Salzberg, S., Shao, W., Shue, B., Sun, J., Wang, Z., Wang, A., Wang, X., Wang, J., Wei, M., Wides, R., Xiao, C., Yan, C., Yao, A., Ye, J., Zhan, M., Zhang, W., Zhang, H., Zhao, Q., Zheng, L., Zhong, F., Zhong, W., Zhu, S., Zhao, S., Gilbert, D., Baumhueter, S., Spier, G., Carter, C., Cravchik, A., Woodage, T., Ali, F., An, H., Awe, A., Baldwin, D., Baden, H., Barnstead, M., Barrow, I., Beeson, K., Busam, D., Carver, A., Center, A., Cheng, M. L., Curry, L., Danaher, S., Davenport, L., Desilets, R., Dietz, S., Dodson, K., Doup, L., Ferriera, S., Garg, N., Gluecksmann, A., Hart, B., Haynes, J., Haynes, C., Heiner, C., Hladun, S., Hostin, D., Houck, J., Howland, T., Ibegwam, C., Johnson, J., Kalush, F., Kline, L., Koduru, S., Love, A., Mann, F., May, D., McCawley, S., McIntosh, T., McMullen, I., Moy, M., Moy, L., Murphy, B., Nelson, K., Pfannkoch, C., Pratts, E., Puri, V., Qureshi, H., Reardon, M., Rodriguez, R., Rogers, Y. H., Romblad, D., Ruhfel, B., Scott, R., Sitter, C., Smallwood, M., Stewart, E., Strong, R., Suh, E., Thomas, R., Tint, N. N., Tse, S., Vech, C., Wang, G., Wetter, J., Williams, S., Williams, M., Windsor, S., Winn-Deen, E., Wolfe, K., Zaveri, J., Zaveri, K., Abril, J. F., Guigó, R., Campbell, M. J., Sjolander, K. V., Karlak, B., Kejariwal, A., Mi, H., Lazareva, B., Hatton, T., Narechania, A., Diemer, K., Muruganujan, A., Guo, N., Sato, S., Bafna, V., Istrail, S., Lippert, R., Schwartz, R., Walenz, B., Yooseph, S., Allen, D., Basu, A., Baxendale, J., Blick, L., Caminha, M., Carnes-Stine, J., Caulk, P., Chiang, Y. H., Coyne, M., Dahlke, C., Mays, A., Dombroski, M., Donnelly, M., Ely, D., Esparham, S., Fosler, C., Gire, H., Glanowski, S., Glasser, K., Glodek, A., Gorokhov, M., Graham, K., Gropman, B., Harris, M., Heil, J., Henderson, S., Hoover, J., Jennings, D., Jordan, C., Jordan, J., Kasha, J., Kagan, L., Kraft, C., Levitsky, A., Lewis, M., Liu, X., Lopez, J., Ma, D., Majoros, W., McDaniel, J., Murphy, S., Newman, M., Nguyen, T., Nguyen, N., Nodell, M., Pan, S., Peck, J., Peterson, M., Rowe, W., Sanders, R., Scott, J., Simpson, M., Smith, T., Sprague, A., Stockwell, T., Turner, R., Venter, E., Wang, M., Wen, M., Wu, D., Wu, M., Xia, A., Zandieh, A., and Zhu, X. (2001). The sequence of the human genome. *Science*, **291**(5507), 1304–1351.

Walker, B. J., Abeel, T., Shea, T., Priest, M., Abouelliel, A., Sakthikumar, S., Cuomo, C. A., Zeng, Q., Wortman, J., Young, S. K., and Earl, A. M. (2014). Pilon: an inte-

grated tool for comprehensive microbial variant detection and genome assembly improvement. *PLoS One*, **9**(11), e112963.

Walkowiak, S., Gao, L., Monat, C., Haberer, G., Kassa, M. T., Brinton, J., Ramirez-Gonzalez, R. H., Kolodziej, M. C., Delorean, E., Thambugala, D., Klymiuk, V., Byrns, B., Gundlach, H., Bandi, V., Siri, J. N., Nilsen, K., Aquino, C., Himmelbach, A., Copetti, D., Ban, T., Venturini, L., Bevan, M., Clavijo, B., Koo, D.-H., Ens, J., Wiebe, K., N'Diaye, A., Fritz, A. K., Gutwin, C., Fiebig, A., Fosker, C., Fu, B. X., Accinelli, G. G., Gardner, K. A., Fradgley, N., Gutierrez-Gonzalez, J., Halstead-Nussloch, G., Hatakeyama, M., Koh, C. S., Deek, J., Costamagna, A. C., Fobert, P., Heavens, D., Kanamori, H., Kawaura, K., Kobayashi, F., Krasileva, K., Kuo, T., McKenzie, N., Murata, K., Nabeka, Y., Paape, T., Padmarasu, S., Percival-Alwyn, L., Kagale, S., Scholz, U., Sese, J., Juliana, P., Singh, R., Shimizu-Inatsugi, R., Swarbreck, D., Cockram, J., Budak, H., Tameshige, T., Tanaka, T., Tsuji, H., Wright, J., Wu, J., Steuernagel, B., Small, I., Cloutier, S., Keeble-Gagnère, G., Muehlbauer, G., Tibbets, J., Nasuda, S., Melonek, J., Hucl, P. J., Sharpe, A. G., Clark, M., Legg, E., Bharti, A., Langridge, P., Hall, A., Uauy, C., Mascher, M., Krattinger, S. G., Handa, H., Shimizu, K. K., Distelfeld, A., Chalmers, K., Keller, B., Mayer, K. F. X., Poland, J., Stein, N., McCartney, C. A., Spannagl, M., Wicker, T., and Pozniak, C. J. (2020). Multiple wheat genomes reveal global variation in modern breeding. *Nature*, **588**(7837), 277–283.

Walsh, J. B. and Marks, J. (1986). Sequencing the human genome. *Nature*, **322**(6080), 590–590.

Wang, B., Yang, X., Jia, Y., Xu, Y., Jia, P., Dang, N., Wang, S., Xu, T., Zhao, X., Gao, S., Dong, Q., and Ye, K. (2021). High-quality arabidopsis thaliana genome assembly with nanopore and HiFi long reads. *Genomics Proteomics Bioinformatics*.

Wang, Y., van der Hoeven, R. S., Nielsen, R., Mueller, L. A., and Tanksley, S. D. (2005). Characteristics of the tomato nuclear genome as determined by sequencing undermethylated EcoRI digested fragments. *Theor. Appl. Genet.*, **112**(1), 72–84.

Wick, R. R., Schultz, M. B., Zobel, J., and Holt, K. E. (2015). Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics*, **31**(20), 3350–3352.

Wicker, T., Sabot, F., Hua-Van, A., Bennetzen, J. L., Capy, P., Chalhoub, B., Flavell, A., Leroy, P., Morgante, M., Panaud, O., Paux, E., SanMiguel, P., and Schulman, A. H. (2007). A unified classification system for eukaryotic transposable elements. *Nat. Rev. Genet.*, **8**(12), 973–982.

Wójtowicz, J. and Gieczewska, K. B. (2021). The arabidopsis accessions selection is crucial: Insight from photosynthetic studies. *Int. J. Mol. Sci.*, **22**(18).

Wood, D. E., Lu, J., and Langmead, B. (2019). Improved metagenomic analysis with kraken 2. *Genome Biol.*, **20**(1), 257.

Xin, H., Nahar, S., Zhu, R., Emmons, J., Pekhimenko, G., Kingsford, C., Alkan, C., and Mutlu, O. (2016). Optimal seed solver: optimizing seed selection in read mapping. *Bioinformatics*, **32**(11), 1632–1642.

Xu, G., Ma, H., Nei, M., and Kong, H. (2009). Evolution of f-box genes in plants: different modes of sequence divergence and their relationships with functional diversification. *Proc. Natl. Acad. Sci. U. S. A.*, **106**(3), 835–840.

Ye, K., Schulz, M. H., Long, Q., Apweiler, R., and Ning, Z. (2009). Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, **25**(21), 2865–2871.

Zapata, L., Ding, J., Willing, E.-M., Hartwig, B., Bezdan, D., Jiao, W.-B., Patel, V., Velikkakam James, G., Koornneef, M., Ossowski, S., and Schneeberger, K. (2016). Chromosome-level assembly of arabidopsis thaliana ler reveals the extent of translocation and inversion polymorphisms. *Proc. Natl. Acad. Sci. U. S. A.*, **113**(28), E4052–60.

Zhou, Y., Minio, A., Massonnet, M., Solares, E., Lv, Y., Beridze, T., Cantu, D., and Gaut, B. S. (2019). The population genetics of structural variants in grapevine domestication. *Nat Plants*, **5**(9), 965–979.

Zmienko, A., Marszalek-Zenczak, M., Wojciechowski, P., Samelak-Czajka, A., Luczak, M., Kozlowski, P., Karlowski, W. M., and Figlerowicz, M. (2020). AthCNV: A map of DNA copy number variations in the arabidopsis genome. *Plant Cell*, **32**(6), 1797–1819.