Finding genomic differences from whole-genome assemblies using SyRI

Inaugural-Dissertation

zur

Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität zu Köln

vorgelegt von

Manish Goel

aus Ghaziabad, Uttar Pradesh, India

Köln, 2020

View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Kölner UniversitätsPublikationsServer

Die vorliegende Arbeit wurde am Max-Planck-Institut für Pflanzenzüchtungsforschung in Köln in der Abteilung Chromosomenbiologie (Direktor: Prof. Dr. Raphael Mercier) innerhalb der Arbeitsgruppe von Prof. Dr. Korbinian Schneeberger angefertigt.

MAX-PLANCK-GESELLSCHAFT

Max-Planck-Institut für
Pflanzenzüchtungsforschung

Erster Referent und Prüfer:	Prof. Dr. Korbinian Schneeberger
Zweite Referentin und Prüferin:	Prof. Dr. Achim Tresch
Beisitzerin/Schriftführerin:	Dr. Kristin Krause
Vorsitzende der Prüfungskommission:	Prof. Dr. Thomas Wiehe

Tag der Disputation:	26.06.2020

Table of Contents

Table of Contents v

List of Figures ix

List of Tables xiii

List of Abbreviations xv

Publication xvii

Acknowledgement xix

Abstract xxiii

Zusammenfassung xxv

1 Introduction 29

1.1 Sequence and structural differences in genomes 30

1.2 Effects of structural rearrangement 31

1.3 Using whole-genome assemblies for genomic differences identification 32

2 Results 35

2.1 SyRI: algorithmic description 36

2.1.1 Syntenic region identification 38

2.1.2 Inversion identification 39

2.1.3 Translocation and duplication (TD) identification 39

2.1.4 Annotation block 40

2.1.5 Sequence difference identification 40

2.2 Performance evaluation using simulated genomes 40

2.2.1 Genomic difference identification methods 41

2.2.2 Genome comparisons 41

2.2.3 SyRI accurately identified simulated genomic differences.........43

2.3 Performance evaluation using real genomes 47

2.4 Effect of assembly contiguity on SyRI 50

2.5 Analysing multiple model species using SyRI...................................52

2.6 Example of a highly rearranged region 53

2.7 Experimental validation of predicted translocations..........................54

3 Methods..59

3.1 Syntenic region identification...59

3.2 Inversion identification...60

3.3 Translocation and duplication (TD) identification 61

3.3.1 Overlapping candidates have inter-dependent annotations......62

3.3.2 Candidate TD identification...62

3.3.3 Selecting optimal candidate TDs 63

3.4 Simulation analysis 65

3.4.1 Generating rearranged genomes from the *A. thaliana* reference genome...65

3.4.2 Simulating reads from the rearranged genomes 65

3.4.3 Finding genomic differences using various methods 65

3.4.4 Calculating prediction accuracy 67

3.5 Comparing real genomes and comparison against the gold-standards dataset...67

3.6 Generating homology-based pseudo-chromosome level assemblies68

3.7 Incomplete assemblies generation and analysis 68

3.7.1 Single incomplete assembly 68

3.7.2 Double incomplete assemblies...69

3.8 Aligning whole-genome assemblies 69

3.9 Validation of predicted translocations..69

3.9.1 Test 1: Absence of translocated sequence.....................................70

3.9.2 Test 2: Modelling expected and observed copy numbers 70
3.9.3 Test 3: Genotype derived sample clustering 70
4 Discussion and Conclusion 73
4.1 Whole-genome assembly based genomic differences identification 73
4.1.1 Changing the paradigm of structural rearrangement 74
4.1.2 Genome-wide optimisation for rearrangements......................... 74
4.1.3 Nested sequence differences and their identification................. 75
4.2 Efficient genomic differences identification by SyRI 75
4.3 Current limitations of SyRI 76
4.4 Towards population-level genomic variance identification.............. 77
4.5 Concluding remarks 78
5 Bibliography 81
6 Appendix 89
6.1 Using SyRI and a working example 89
6.1.1 Installing SyRI.. 89
6.1.2 Working example 90
6.2 Additional Tables:.. 92
6.3 Additional Figures 95
7 Erklärung zur Dissertation 111

List of Figures

Figure 1: SyRI's workflow for genomic differences identification. 37

Figure 2: Genomic difference hierarchy..38

Figure 3: Sequence differences classification..41

Figure 4: Structural rearrangement breakpoints...42

Figure 5: Structural rearrangement identification performance comparison.44

Figure 6: Incorrect annotation of transpositions and distal duplications.......45

Figure 7: Indel identification performance comparison. 46

Figure 8: Total size and number of rearrangements and indels identified by different methods in the NA19240 genome..47

Figure 9: Gold-standard genomic differences identification by methods......48

Figure 10: Total size and number of rearrangements and indels identified by different methods in the L*er* genome. 49

Figure 11 Workflow for analyzing the effects of genome contiguity. 50

Figure 12: Rearrangement prediction efficiency of SyRI in incomplete assemblies. 51

Figure 13: Multiple co-occurring rearrangements. 54

Figure 14: Validating translocations using recombination induced copy-number changes...55

Figure S1: SyRI's rearrangement performance when both assemblies are incomplete...95

Figure S2: Genomic differences between human reference genome (Hg38) and genome NA19240. 96

Figure S3: Genomic differences between human reference genome (Hg38) and genome NA12878. 97

Figure S4: Genomic differences between yeast reference genome (S288C) and accession YJM1447. 98

Figure S5: Genomic differences between fruit-fly reference genome (iso-1) and accession A4. 99

Figure S6: Genomic differences between maize reference genome (B73) and accession PH207. 100

Figure S7: Genomic differences between *A. thaliana* reference genome (Col-0) and accession L*er*. 101

Figure S8: Using a Directed Acyclic Graph (DAG) for syntenic path identification. 102

Figure S9: Various conformation of inversions and inverted alignments. 103

Figure S10: Two-step method for inversion identification. 104

Figure S11: Complexities of translocations and duplication identification. 105

Figure S12: Using a Directed Acyclic Graph (DAG) for candidate TD identification. 106

Figure S13: Selecting non-conflicting candidate TDs. 107

Figure S14: Homology based pseudo-chromosome generation. 108

List of Tables

Table 1: Computation resources used and structural differences identified by SyRI for difference genomes. 53

Table S1: Statistics of assemblies used. 92

Table S2: Number of large (>1Mb) variations identified by each method in L*er* genome. 93

Table S3: Whole-genome alignment commands used. 94

List of Abbreviations

bp	basepairs
CNV	Copy-number variation
Col-0	Columbia-0
CPU	Central processing unit
DAG	Directed acyclic graph
DNA	Deoxyribonucleic acid
e.g.	For example
GB	Gigabyte
Gb	Trillion-basepairs
GWAS	Genome-wide association studies
InDel	Insertion and Deletion
Inv	Inversion
Kb	Thousand-basepairs
L*er*	*Landsberg erecta*
MB	Megabyte
Mb	Million-basepairs
PhD	Doctor of Philosophy
SNP	Single nucleotide polymorphism
SNV	Single nucleotide variation
Syn	Synteny
SyRI	Synteny and Rearrangement Identifier
TD	Translocation and duplication
Trans	Translocation

Publication

Goel, M., Sun, H., Jiao, W. *et al*. SyRI: finding genomic rearrangements and local sequence differences from whole-genome assemblies. *Genome Biol* **20,** 277 (2019). https://doi.org/10.1186/s13059-019-1911-0

Acknowledgement

First and foremost, I would like to thank my supervisor Korbinian Schneeberger, Max Planck Institute for Plant Breeding Research (MPIPZ), for providing me with the opportunity to work on such an interesting and meaningful problem. Your guidance and suggestions, both scientific and non-scientific, throughout my PhD, were invaluable. I would also like to thank you for creating a stimulating work environment in the group, where doing my PhD was more fun and less work.

I would also like to thank all of my colleagues in the Schneeberger group: Hequan Sun, Wen-Biao Jiao, Vidya Oruganti, Jose-Antonio Campoy, Kristin Krause and Onur Dogan. Special thanks to Hequan, my office-mate for the last 40 months, for always be willing to share his experience and expertise with me. I am thankful to all of you for your support and for creating such a friendly group. Indeed, our table-tennis matches will always remain a fond memory for me. I would also like to thank George Coupland (Director, Department of Plant Developmental Biology) and Raphael Mercier (Director, Department of Chromosome Biology) for hosting me in their departments, Angela Hancock and Xiangchao Gan for giving important feedbacks during my TAC meetings. I am also thankful to Stephan Wagner (IMPRS Coordinator) and Marc Thoben (IT Services, MPIPZ) who were extremely helpful during various administrative and technical problems I encountered during my PhD.

Special thanks to my parents who always supported me and made me move forward without worrying about life and to Anshupa Sahu for always being there for me. Extensive gratitude to all my friends and family for their constant support.

Finally, I would like to thank me who actually worked and wrote this thesis and you *The Reader* for showing your interest in reading this work.

Dedicated to India, my country, an idea, a dream where everyone would be equal and everyone would be free. I hope that this dream stays safe, forever.

(inspired by Aamir Aziz)

Abstract

Genomic differences can range from single nucleotide differences (SNPs) to large complex structural rearrangements. Current methods typically can annotate sequence differences like SNPs and large indels accurately but do not unravel the full complexity of structural rearrangements that include inversions, translocations, and duplications. Structural rearrangements involve changes in location, orientation, or copy-number between highly similar sequences and have been reported to be associated with several biological differences between organisms. However, they are still scantly studied with sequencing technologies as it is still challenging to identify them accurately.

Here I present SyRI, a novel computational method for genome-wide identification of structural differences using the pairwise comparison of whole-genome chromosome-level assemblies. SyRI uses a unique approach where it first identifies all syntenic (structurally conserved) regions between two genomes. Since all non-syntenic regions are structural rearrangements by definition, this transforms the difficult problem of rearrangement identification to a comparatively easier problem of rearrangement classification. SyRI analyses the location, orientation, and copy-number of alignments between rearranged regions and selects alignments that best represent the putative rearrangements and result in the highest total alignment score between the genomes. Next, SyRI searches for sequence differences that are distinguished for residing in syntenic or rearranged regions. This distinction is important, as rearranged regions (and sequence differences within them) do not follow Mendelian Law of Segregation and are therefore inherited differently compared to syntenic regions. Using SyRI, I successfully identified rearrangements in human, *A. thaliana*, yeast, fruit fly, and maize genomes. Further, I also experimentally validated 92% (108/117) of the predicted translocations in *A. thaliana* using a genetic approach.

Zusammenfassung

Genomische Unterschiede können von Einzelnukleotidunterschieden (SNPs) bis zu großen komplexen strukturellen Variationen reichen. Gegenwärtige Verfahren können typischerweise Sequenzunterschiede wie SNPs und große Indels genau annotieren, aber nicht die volle Komplexität struktureller Umlagerungen aufdecken, die Inversionen, Translokationen und Duplikationen umfassen. Strukturelle Umlagerungen beinhalten Änderungen der Position, Orientierung oder Kopienzahl zwischen sehr ähnlichen Sequenzen und es wurde berichtet, dass sie mit mehreren biologischen Unterschieden zwischen Organismen verbunden sind. Sie werden jedoch immer noch kaum mit Sequenzierungstechnologien untersucht, da es immer noch schwierig ist, sie genau zu identifizieren.

Hier präsentiere ich SyRI, eine neuartige Berechnungsmethode zur genomweiten Identifizierung von Strukturunterschieden unter Verwendung des paarweisen Vergleichs von Chromosomen-Level-Assemblies im gesamten Genom. SyRI verwendet einen einzigartigen Ansatz, bei dem zunächst alle syntenischen (strukturell konservierten) Regionen zwischen zwei Genomen identifiziert werden. Da alle nicht syntenischen Regionen per Definition strukturelle Umlagerungen sind, wandelt dies das schwierige Problem der Identifizierung von Umlagerungen in ein vergleichsweise einfacheres Problem der Klassifizierung von Umlagerungen um. SyRI analysiert die Position, Orientierung und Kopienzahl der Alignments zwischen neu angeordneten Regionen und wählt Alignments aus, die die mutmaßlichen Umlagerungen am besten darstellen und zu der höchsten Gesamtausrichtungsbewertung zwischen den Genomen führen. Als nächstes sucht SyRI nach Sequenzunterschieden, die für den Aufenthalt in syntenischen oder neu angeordneten Regionen unterschieden werden. Diese Unterscheidung ist wichtig, da neu angeordnete Regionen (und Sequenzunterschiede innerhalb dieser) nicht dem Mendelschen Segregationsgesetz folgen und daher anders

vererbt werden als syntenische Regionen. Mit SyRI konnte ich erfolgreich Umlagerungen in Genomen von Menschen, A. thaliana, Hefen, Fruchtfliegen und Mais identifizieren. Außerdem habe ich 92% (108/117) der vorhergesagten Translokationen in A. thaliana unter Verwendung eines genetischen Ansatzes experimentell validiert.

1 Introduction

Parts of this chapter were the basis of a manuscript that was published as a peer-reviewed research article in Genome Biology (Goel et al. 2019)[1]*. However, in this chapter, I only discuss the work that was done by me. Results and data that were not generated by me are either clearly pointed or cited.*

Authors list (Goel et al. 2019): Manish Goel[α]*, Hequan Sun*[β]*, Wen-Biao Jiao*[α]*, Korbinian Schneeberger*[α, β]*.*

Author affiliations: [α]*Department of Chromosome Biology, Max Planck Institute for Plant Breeding Research, 50829 Cologne, Germany;* [β]*LMU Munich, 82152 Planegg-Martinsried, Germany.*

Authors contributions (Goel et al. 2019): The project was conceived by KS and WBJ. MG and KS developed the algorithms. MG implemented SyRI and performed all analyses. HS processed recombinant genome sequencing data and identified crossing-over sites. WBJ generated the Ler assembly. The manuscript was written by MG and KS with inputs from HS and WBJ. All authors read and approved the final manuscript.

Genomes serve as the central data storage hubs for storing biological information for all known life forms on earth[2]. They are composed of one or more long polymers of deoxyribonucleic acid (DNA). These molecules are called chromosomes. Each chromosome contains multiple genes that are the representative unit of genomic information and are used for the synthesis of proteins, the building blocks for all living organisms[3]. Often, the genomic sequence of an organism can acquire a spontaneous error called mutation that makes it different from other individuals of the same species. Though mostly benign, these differences can result in fatal diseases or provide evolutionary advantages. Besides recombination of existing genomes during sexual reproduction, accumulation of such mutations in genomes is a critical driving force for the evolution of species making genomic differences one of the primary sources of the observed biological diversity[4]. Consequently, to better understand and explore the natural differences present among various species as well as among individuals of a species, it is important to identify and study the differences present in their genomes.

1.1 Sequence and structural differences in genomes

Genomic differences can be classified as *differences in sequence* and *differences in structure*. Differences in sequence are variation in locally conserved regions on the genome and do not involve relocation of genomic regions. Consequently, differences in sequence do not affect genome collinearity. These include single nucleotide variation (SNVs), insertions and deletions (indels), or structural variation (e.g. large indels, tandem repeats). Differences in structure are variation where genomic regions change in the location, orientation, or copy-number. As these differences alter the order and orientation of genomic regions, thus disrupting genomic collinearity, they are not structurally conserved and are collectively referred to as structural rearrangements. These include inversions, translocations, and duplications. Inversions are variation when a region gets reverse complemented at its locus. Translocations involve relocation of genomic regions, whereas duplications involve copying of regions thus changing their copy-number. Translocated and duplicated regions are often also reverse complemented resulting in inverted translocations and duplications, respectively.

Further, translocations can be intra-chromosomal transpositions (relocation to a different region in the same chromosome) or inter-chromosomal translocations (relocation to a different chromosome). Similarly, duplications can be tandem-duplications (addition of a duplicated copy adjacent to the original) or distal-duplications (duplication to a different location, also known as segmental duplications). However, for simplicity, unless specified otherwise, I would consider transpositions and translocations together as ‘translocations’; and tandem and distal duplications together as ‘duplications’.

1.2 Effects of structural rearrangement

Structural rearrangements represent more variation in genomes compared to single nucleotide polymorphisms (SNPs) and often disrupt functionally relevant regions (like genes) resulting in phenotypic differences[5]. Indeed, it has been shown that large rearrangements have a more significant effect on expression compared to SNPs and indels[6]. Further, as structural rearrangements disrupt collinearity, recombination is infrequent in these regions[7]. Over time, random mutations can accumulate in rearranged regions leading to population stratification[8]. Individuals with such rearrangements could express different phenotypes and with the continuous accumulation of genomic differences can evolve into new species.

In humans, structural rearrangements are associated with multiple diseases[9]. A recent study demonstrated that structural rearrangements may lead to inactivation of tumour suppressors genes while also activating cancer driver genes[10]. Structural rearrangements are also associated with misexpression of genes resulting in limb malformation syndromes[11]. Multiple neurological diseases have also been found to be associated with structural rearrangements including autism, schizophrenia, and bipolar disorder[12–14].

Structural rearrangements could also have evolutionary effects[15]. For example, when humans started doing agriculture and thus consumed more starch, then duplication in salivary amylase gene (catalyst for hydrolysis of starch to sugar) was positively selected[16]. Structural rearrangements were also involved in the development of antifolate (drugs used in malaria treatment) resistant *Plasmodium* falciparum parasite[17]. Similar observations have been made in fruit fly as well, where rearrangements in toxin-response genes were found to be under positive-selection[18]. In plants, R-gene clusters are known to be hotspots of rearrangements which helps in the development of resistance[19]. A duplication of the RCO gene in *Cardamine hirsute* resulted in increased leaf shape complexity. Further, loss of the same gene leads to simpler leaves in *Arabidopsis thaliana*[20].

Structural rearrangements are also important for breeding research. In tomato, a structural rearrangement was used to improve breeding efficiency[21]. In dog breeding, structural rearrangements are being studied as a source for the differences across various breeds[22].

1.3 Using whole-genome assemblies for genomic differences identification

Many of the current genomic differences identification methods utilize sequencing reads for genomic differences identification. The reads are aligned to the reference genome sequence and the alignment breakpoints are processed for identifying genomic differences[23]. This approach can identify sequence differences (like SNPs, indels, and structural variation) with high accuracy; however, accurate prediction of structural rearrangements remains challenging. In contrast, whole-genome assemblies are considered as the gold-standard data for the identification of all rearrangements as assembled sequences are typically much longer and of higher quality as compared to raw sequence reads[24]. However, despite recent technological improvements in methods for generation of whole-genome *de novo* assemblies[25], there are so far only a few methods that can identify genomic differences from whole-genome assemblies[26]. Available methods include AsmVar, Assemblytics, and smartie-sv[27–29]. AsmVar compares individual sequences (scaffolds/contigs) from the query genome assembly against the reference sequence and analyses alignment breakpoints to identify inversions and translocations[27]. Assemblytics utilizes uniquely aligned regions within contigs from the query genome and the reference sequence to identify genomic differences like large indels and local repeats[28]. Smartie-sv too identifies differences by comparing individual alignments between query genome assembly and reference genome[29].

In this thesis, I introduce **SyRI (Synteny and Rearrangement Identifier)**, a novel computational method for the identification of all genomic differences. SyRI uses whole-genome alignments generated by aligning two chromosome-level whole-genome assemblies as input and identifies structurally conserved (syntenic) and rearranged regions in the two genomes. Afterwards, SyRI also identifies local sequence differences within all syntenic as well as rearranged regions. Noteworthily, SyRI provides complete regional annotation of rearrangements by identifying coordinates of genomic differences in both genomes, thus reporting which region in reference was rearranged and where that rearranged region is located in the query genome. This is a significant improvement compared to current methods that typically do not annotate breakpoints for all rearrangements in both genomes[30–32]. Additionally, current methods have limited functionality in identifying transpositions and distal duplications. Thus, SyRI is the first method that can accurately identify all classes of rearrangements, including transpositions and distal duplications.

Finally, I analysed SyRI's performance and compared it with current methods using simulated rearranged genomes as well as gold-standard genomic differences data. Using SyRI, I identified genomic differences in divergent genomes of five model species. This also included two *A. thaliana* strains, for which I experimentally validated over 100 predicted translocations using a genetic approach.

2 Results

Parts of this chapter were the basis of a manuscript that was published as a peer-reviewed research article in Genome Biology (Goel et al. 2019)[1]*. However, in this chapter, I only discuss the work that was done by me. Results and data that were not generated by me are either clearly pointed or cited.*

Authors list (Goel et al. 2019): Manish Goel[α]*, Hequan Sun*[β]*, Wen-Biao Jiao*[α]*, Korbinian Schneeberger*[α, β]*.*

Author affiliations: [α]*Department of Chromosome Biology, Max Planck Institute for Plant Breeding Research, 50829 Cologne, Germany;* [β]*LMU Munich, 82152 Planegg-Martinsried, Germany.*

Authors contributions (Goel et al. 2019): The project was conceived by KS and WBJ. MG and KS developed the algorithms. MG implemented SyRI and performed all analyses. HS processed recombinant genome sequencing data and identified crossing-over sites. WBJ generated the Ler assembly. The manuscript was written by MG and KS with inputs from HS and WBJ. All authors read and approved the final manuscript.

During my PhD, I developed a novel computational method for the identification of genomic differences between two closely related individuals. This method called “SyRI” for **Sy**nteny and **R**earrangement **I**dentifier is, to the best of my knowledge, the first method that identifies all classes of structural rearrangements. SyRI identifies genomic differences by analyzing whole-genome alignments between chromosome-level assemblies. Additionally, SyRI is also the first method that annotates sequence differences within structurally rearranged regions.

In this chapter, I present the novel methodologies applied in SyRI and describe the various steps involved in the identification of all genomic differences. I also discuss

SyRI's performance in simulated as well as gold standard datasets and its comparison with currently popular methods. Since it is not always possible to have chromosome-level assemblies; I also describe SyRI's usefulness in genomic differences identification using homology-based pseudo-chromosome level assemblies. I also demonstrate SyRI's usability in finding genomic differences by analyzing genomes of five model species and present an example of a highly rearranged region consisting of large structural rearrangements overlapping with multiple genes highlighting the importance of efficient genomic difference identification. Finally, I describe the validation of SyRI's prediction using a population of 50 F2 plants generated by hybridising the Col-0 and L*er* accessions (strains) of *A. thaliana*.

2.1 SyRI: algorithmic description

SyRI identifies genomic differences between two chromosome-level assemblies by analyzing the whole-genome alignments between them (**Figure 1**). It starts by identifying the longest syntenic regions between homologous chromosomes in the genomes. Synteny refers to structurally conserved regions as they are collinear (having same relative position) in the homologous chromosomes. By extension, this suggests that all non-syntenic regions are structurally rearranged. Therefore, identification of syntenic regions simultaneously also identifies structural rearrangements (**Figure 1**: Step 1). This novel approach transforms the difficult problem of structural rearrangement identification to a comparatively easier problem of structural rearrangement annotation. SyRI annotates rearranged regions as inversions, translocations, and duplications. Inversions are similar to syntenic regions as they do not involve the relocation of genomic regions; therefore, they are easier to identify and are annotated next. Finally, SyRI annotates intra-chromosomal transpositions and duplications followed by inter-chromosomal translocations and duplications (**Figure 1**: Step 2).

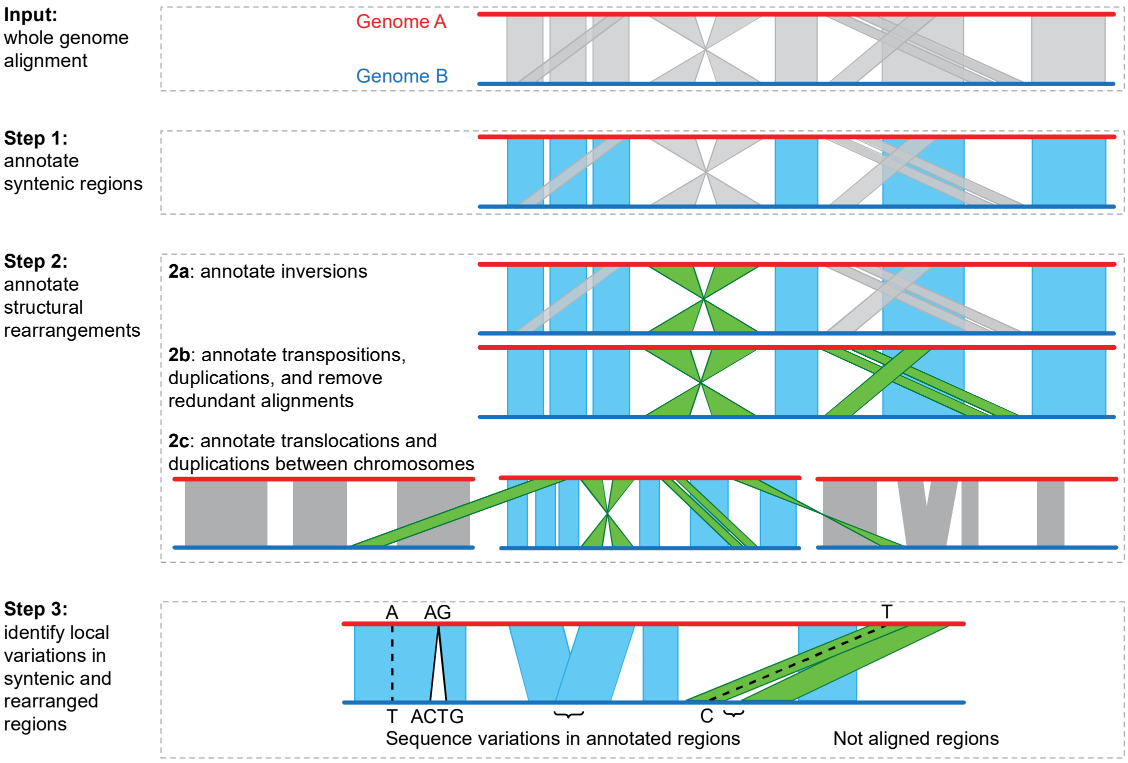

Input:
whole genome alignment
Genome A
Genome B
Step 1:
annotate syntenic regions
Step 2:
annotate structural rearrangements
2a: annotate inversions
2b: annotate transpositions, duplications, and remove redundant alignments
2c: annotate translocations and duplications between chromosomes
Step 3:
identify local variations in syntenic and rearranged regions
A
AG
T
T
ACTG
C
Sequence variations in annotated regions
Not aligned regions

Figure 1: SyRI's workflow for genomic differences identification. SyRI uses whole-genome alignment data as input. Grey polygons represent local alignments between two regions of the genomes. In the Step 1, SyRI identifies syntenic regions (longest collinear region) between homologous chromosomes. In the Step 2 (a-c), SyRI annotates structurally rearranged regions as inversions, transpositions and intra-chromosomal duplications, and translocations and inter-chromosomal duplications. Redundant alignments are filtered out. In the Step 3, SyRI identifies local sequence differences in all syntenic and rearranged regions by analyzing the individual alignments and gaps/overlaps between adjacent alignments of the annotation blocks. SyRI also reports not-aligning regions between annotation blocks. (from Goel *et al.*, 2019)

After annotating syntenic and rearranged regions, SyRI identifies sequence differences (**Figure 1**: Step 3). Noteworthily, SyRI identifies sequence differences in both syntenic as well as rearranged regions. This is important because rearranged regions and sequence differences within them do not follow Mendelian segregation. Further, sequence differences in inverted regions are strongly linked because of lack of recombination in inversions[7], whereas, recombination between the two loci of a transposition or intra-

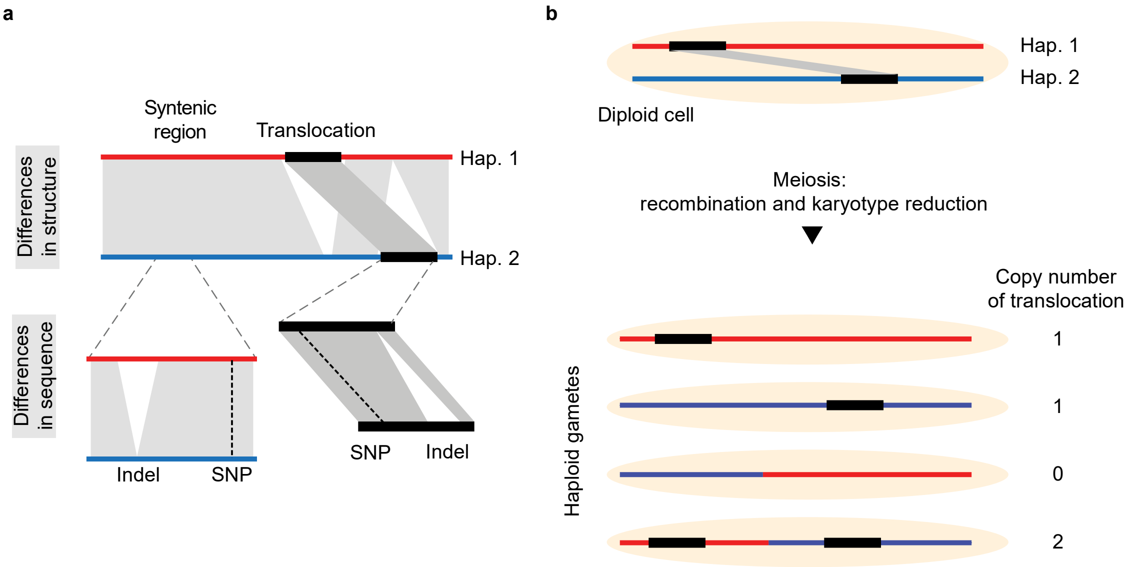

a
Syntenic region
Translocation
Hap. 1
Hap. 2
Differences in structure
Differences in sequence
Indel
SNP
SNP
Indel
b
Hap. 1
Hap. 2
Diploid cell
Meiosis:
recombination and karyotype reduction
Copy number of translocation
Haploid gametes
1
1
0
2

Figure 2: Genomic difference hierarchy. a) Genomes can have differences in structure as well as differences in sequence within the structurally conserved and rearranged regions. b) Meiotic recombination between rearranged loci results in copy-number variation in the haploid gametes. (from Goel *et al.*, 2019)

chromosomal duplication can result in copy-number changes for the rearranged region (**Figure 2**). As a result, sequence differences in rearranged regions can lead to false signals in genome-wide association studies (GWAS), selection screens, as well as recombination analysis[33,34]. SyRI provides an efficient approach to filter out such SNPs and sequence differences by providing a hierarchy of variation where SNPs/indels within structurally rearranged regions are also reported.

2.1.1 Syntenic region identification

Syntenic region identification involves the selection of the largest set of collinear alignments. Starting from whole-genome alignments between homologous chromosomes, SyRI selects all directed alignments between them. Using these alignments, SyRI generates a directed acyclic graph (DAG) where each node corresponds to an alignment and an edge exists between two nodes when they are collinear to each other (Methods: 3.1). SyRI then uses dynamic programming to find the longest path in this graph. This path represents the largest set of collinear alignments that constitute the syntenic region between a pair of homologous chromosomes. This approach is similar to the algorithms used by MUMmer to find whole-genome alignments[35,36].

2.1.2 Inversion identification

An inversion constitutes of one or more inverted alignments that together represent one inversion event. To identify inversions, SyRI selects all inverted alignments between a pair of homologous chromosomes and reverse-complements the query genome to transform inverted alignments into directed alignments. Similar to syntenic identification, SyRI again generates a directed acyclic graph (DAG) using these directed alignments (Methods: 3.2). In this graph, each path corresponds to a putative candidate inversion between two homologous chromosomes, thus, providing all possible inversions. These candidate inversions can overlap and intersect each other resulting in conflicting annotations. To resolve this, SyRI generates a new DAG using candidate inversions and selects candidates that can co-exist in the homologous chromosomes and result in the highest combined alignment score for the inversions and syntenic regions.

2.1.3 Translocation and duplication (TD) identification

After selecting syntenic and inverted regions, SyRI annotates remaining alignments as TDs or removes them as redundant (repetitive). For this, it first identifies transpositions and intra-chromosomal duplications between homologous chromosomes followed by translocations and inter-chromosomal duplications identification from non-homologous chromosomes. Both, intra- and inter-chromosomal differences are identified using the same methodology (Methods: 3.3).

For TD identification, SyRI groups alignments such that each group represents one putative candidate TD. Each candidate TD gets a score based on the length of the individual alignments and the gaps between them (Methods: 3.3.2). SyRI filters out candidates with a low score or high overlap with syntenic or inverted regions. Rearrangements in genomic repeats result in alignments to different copies of the same repeat. This results in the selection of multiple candidate TDs corresponding to the same rearranged region. Similar to inversions, these overlapping candidate TDs result in conflicting annotations. SyRI uses optimization strategies to select a non-conflicting set of candidate TDs while maximizing the total length of the annotated sequence and the total alignment score for the genome (Methods: 3.3.3).

2.1.4 Annotation block

SyRI groups syntenic and rearranged alignments to generate annotation blocks. Each annotation block consists of consecutive alignments that together represent one genomic structural unit. For example, a syntenic block would consist of consecutive and uninterrupted syntenic alignments. Similarly, alignments that together constitute one inversion (or TD) would form an inversion (or TD) block.

2.1.5 Sequence difference identification

SyRI identifies sequence differences (SNPs and small indels) from the aligned sequence in the annotation blocks. Larger structural variation (like CNVs and indels) are identified by comparing the gaps and overlaps between the consecutive alignments within an annotation block (**Figure 3**) similar to the structure variation identification methodology used by Assemblytics[28]. SyRI also reports all un-aligned sequences that are regions between neighbouring annotation blocks but are not part of any annotation block. These regions can be considered as insertions and deletions in their respective genomes, but their corresponding coordinate in the other genome cannot be described.

2.2 Performance evaluation using simulated genomes

To assess SyRI's genomic differences identification performance, I performed a simulation analysis. Using the *A. thaliana* (Col-0) reference genome as a template, I

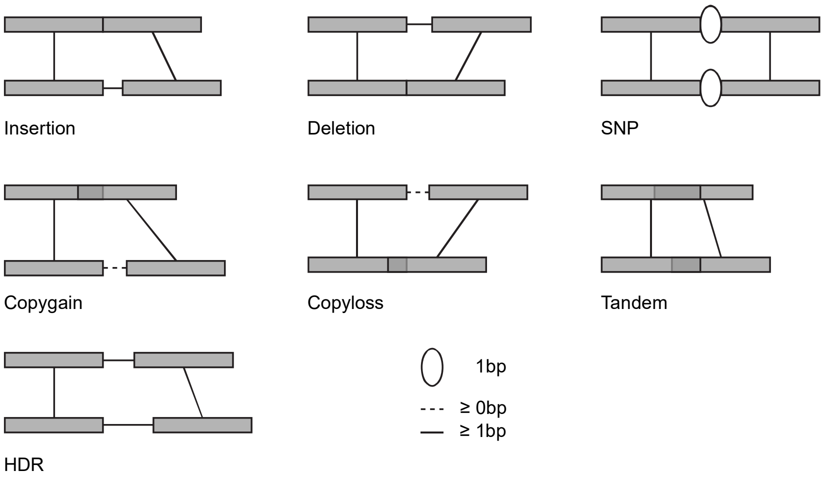


Figure 3: Sequence differences classification. Grey blocks are alignments. Adjacent alignments can be overlapping or can have gaps in-between (shown as dashed line, solid line, and ellipses). (from Goel *et al.*, 2019)

simulated 100 rearranged genomes each consisting of inversions, transpositions, translocations, tandem duplications, distal duplications, or indels, thus, generating 600 rearranged genomes in total (Methods: 3.4.1, **Table S1**). I identified genomic differences from these rearranged genomes using SyRI and other currently available methods and compared their performances.

2.2.1 Genomic difference identification methods

To have a comprehensive comparison, I selected six of the currently available genomic differences identification methods. Among these, three methods were based on whole-genome assemblies (Assemblytics[28], Smartie-SV[29], and AsmVar[27]), two methods required long-read sequencing data (Sniffles[30] and Picky[31]), and one method used short-read sequencing data (LUMPY[32]). For sequencing-reads based methods, I simulated reads from the rearranged genomes: PacBio and Oxford Nanopore reads for long-reads based method and Illumina reads for short-read based method (Method: 3.4.2). I used rearranged genomes directly for assembly-based methods.

2.2.2 Genome comparisons

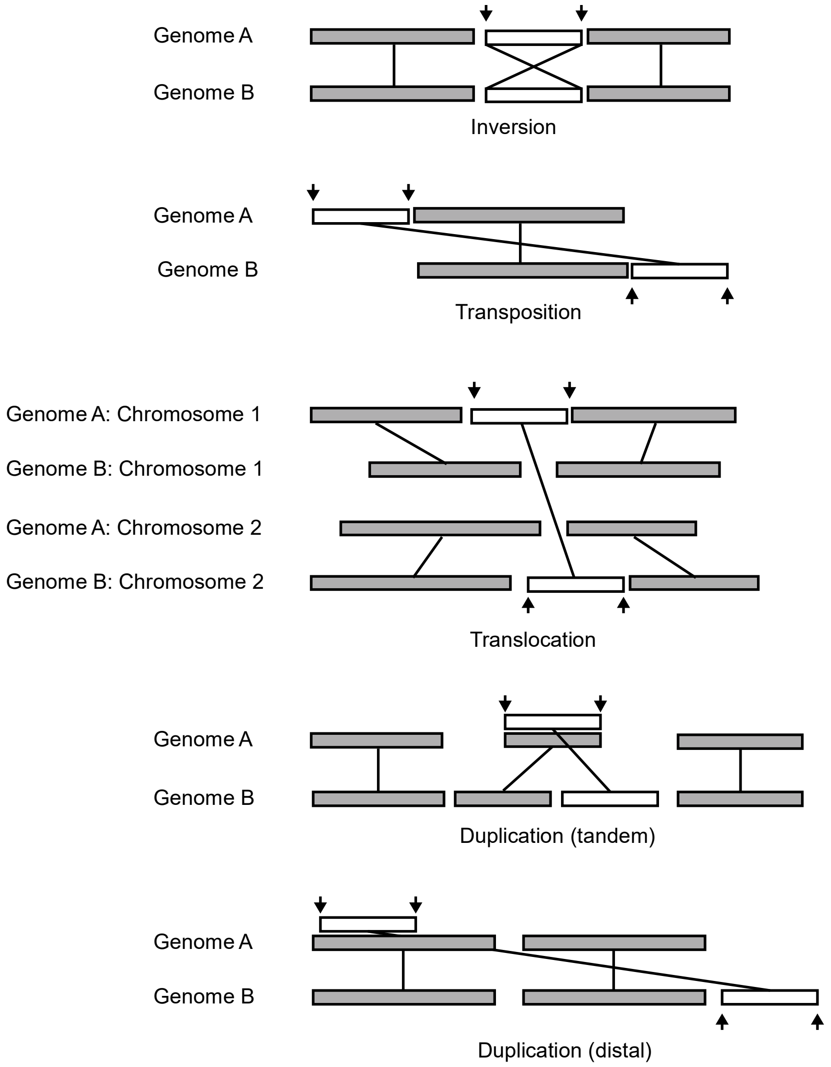

Genome A
Genome B
Inversion
Genome A
Genome B
Transposition
Genome A: Chromosome 1
Genome B: Chromosome 1
Genome A: Chromosome 2
Genome B: Chromosome 2
Translocation
Genome A
Genome B
Duplication (tandem)
Genome A
Genome B
Duplication (distal)

Figure 4: Structural rearrangement breakpoints. Grey alignments represents structurally conserved regions, whereas white alignments represents rearranged regions. To analyse the validity of a predicted rearrangement, coordinates marked with the black arrows were tested against those of the simulated rearrangements. (from Goel *et al.*, 2019)

I aligned all rearranged genomes and the corresponding sequencing reads to Col-0 reference genome and identified genomic differences with all methods (Methods: 3.4.3). I assessed current assembly-based methods for only those rearrangements that they can identify (as they are not designed to identify all types of structural rearrangements). Different methods identify and report rearrangements differently. To ensure a standardized comparison of these methods, inspired by an earlier study[30], I compared the breakpoints of the predicted rearrangements and used correctness classes for the predictions (**Figure 4**).

If a method correctly predicted all breakpoints of a rearrangement together and provided correct annotation, then the method 'identified' the rearrangement. When a method could predict at least one breakpoint with correct annotation, then it 'indicated' the rearrangement. If a method could predict a breakpoint but had the wrong annotation, then the method had 'incorrect' prediction for the rearrangement. Finally, when a method could not predict any breakpoint, then the method 'missed' the rearrangement. For rearrangements involving relocation of the genomic regions (transpositions, translocations, and distal duplications), I checked breakpoints in both reference and query genome. For rearrangement involving modifications at the same loci (inversion and tandem duplications), I checked the coordinates in only the reference genome (**Figure 4**). To compare indel identification performance, I compared the location and size of the simulated and the predicted indels (Method: 3.4.4).

2.2.3 SyRI accurately identified simulated genomic differences

SyRI consistently identified most of the simulated variation for all classes of structural differences (**Figure 5**). Other assembly-based methods were limited by design, as they do not identify all classes of rearrangements. AsmVar could identify inversions and translocations, whereas Smartie-sv and Assemblytics could identify only inversions and duplications respectively. In this simulation analysis, AsmVar accurately identified transpositions and translocations but had incorrect annotations for the majority of the inversions. Assemblytics performed well for the identification of tandem duplications but missed most of the distal duplications. Smartie-sv did not perform well for inversion

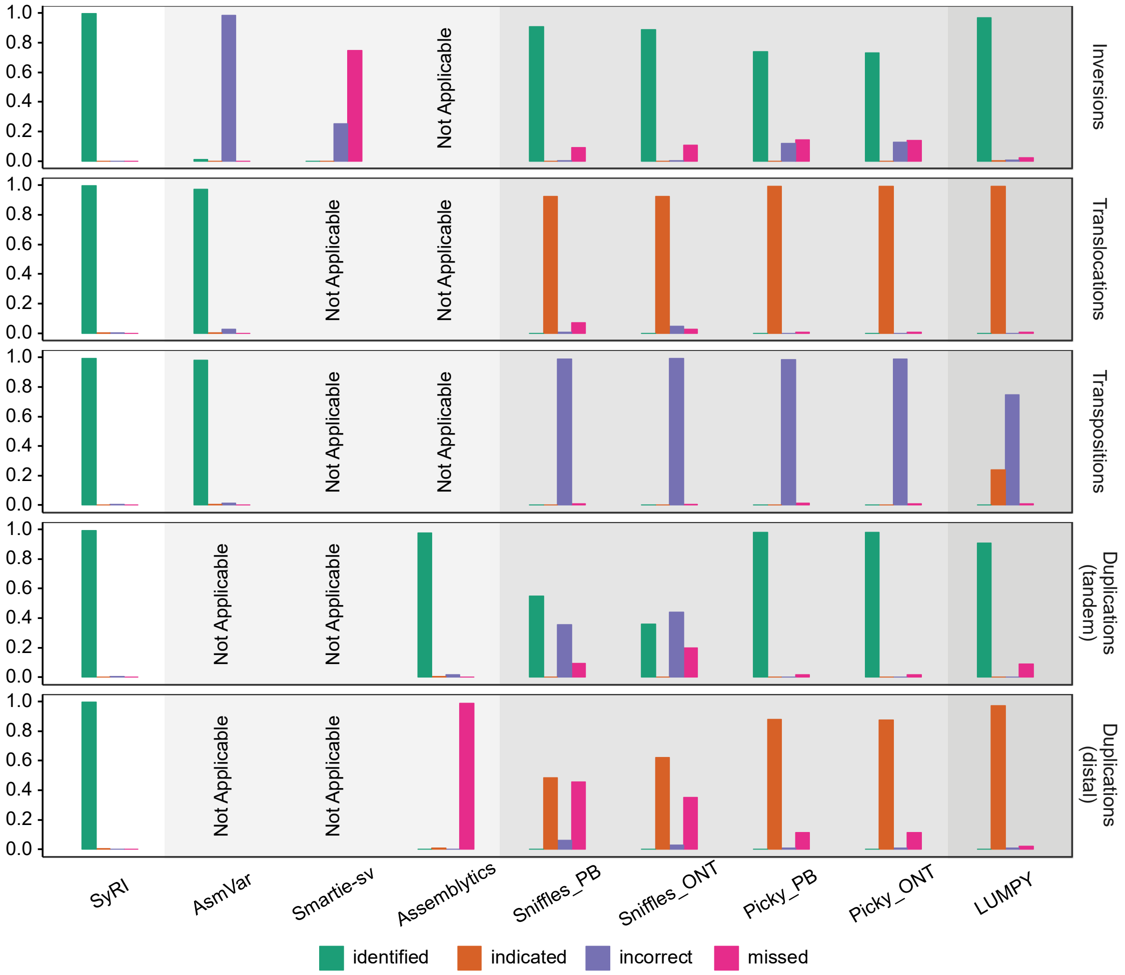


Figure 5: Structural rearrangement identification performance comparison. Vertical bars show the ratio of predicted rearrangements belonging to a specific class. 'Not Applicable' implies that the method is not designed to identify the specific genomic difference. Background colours represent the data type required by the respective methods (from white to dark grey: chromosome-level *de novo* assembly, *de novo* assembly, long sequencing reads (both PacBio (PB) and Oxford Nanopore (ONT) reads), short sequencing reads). (from Goel *et al.*, 2019)

identification and missed most of them. These results suggest that compared to current assembly-based methods, SyRI performed better as it identified all classes of rearrangements as well as identified each class of rearrangement more precisely than current methods.

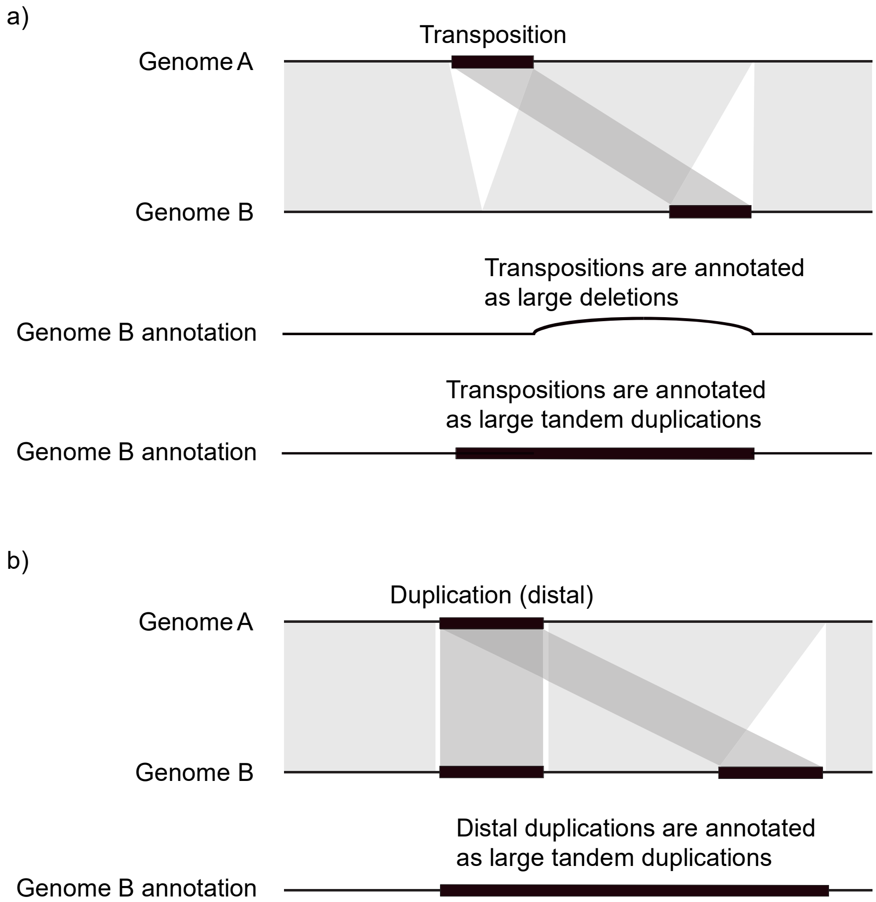

a)
Transposition
Genome A
Genome B
Transpositions are annotated as large deletions
Genome B annotation
Transpositions are annotated as large tandem duplications
Genome B annotation
b)
Duplication (distal)
Genome A
Genome B
Distal duplications are annotated as large tandem duplications
Genome B annotation

Figure 6: Incorrect annotation of transpositions and distal duplications. a) Reads originating from the translocated loci in genome B align to the reference genome similarly to reads originating from deletions and tandem duplications. As a result, read-based methods wrongly annotate transpositions as large deletions and tandem duplications. b) Similarly, read-based methods cannot differentiate between reads originating from large tandem duplications and intra-chromosomal distal duplications; again resulting in wrong annotations. (from Goel *et al.*, 2019)

In contrast, and somewhat counterintuitive, read-based methods performed better compared to current assembly-based methods. All read based methods identified majority of the inversions, while Picky and LUMPY identified the majority of tandem duplications as well. However, they could not identify all breakpoints for transpositions, translocations, and distal duplications. These rearrangements involve genomic region relocation. Since sequencing-reads do not provide information about the query genome

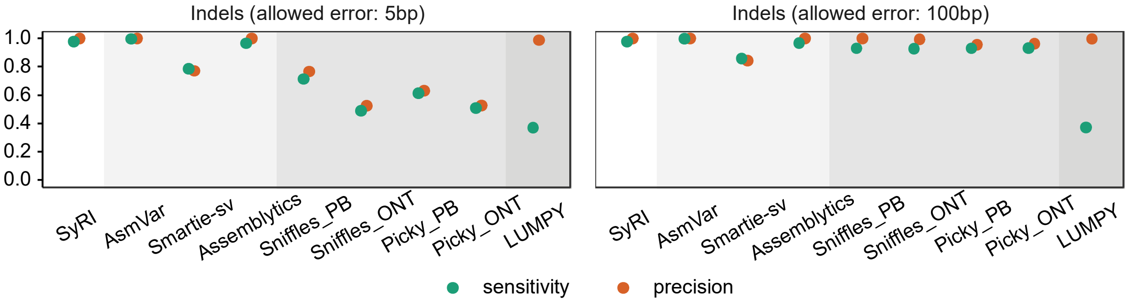


Figure 7: Indel identification performance comparison. Sensitivity (green points) and precision (orange points) values for the prediction of indels by different methods. Y-axis value corresponds to the average performance from 100 simulated genomes and the results are shown for two allowed error values: 5 and 100 bps (Methods: 3.4.4). Background colours represent the data type required by the respective methods (from white to dark grey: chromosome-level de novo assembly, de novo assembly, long sequencing reads (both PacBio (PB) and Oxford Nanopore (ONT) reads), short sequencing reads). (from Goel *et al.*, 2019)

structure; these methods could not identify breakpoints in the query genome for relocated regions. Additionally, reads originating from relocated regions align to the reference genome similarly to reads originating from large deletions or tandem duplications (**Figure 6**). Consequently, read-based methods falsely predicted multiple large deletions and tandem duplications (between homologous chromosomes) that were overlapping transpositions and distal duplications. This resulted in lower performance in rearrangements identification and over-estimation of deletion and tandem duplications in the genome.

All methods had few false-positives for inversions, translocations, and tandem duplications identification. However, for distal duplications, read-based methods and Assemblytics had a high false-positive rate. Whereas Picky and LUMPY had multiple falsely predicted breakpoints for genomes with simulated transpositions.

For indel identification, all assembly-based methods performed better than read-based methods, and the performance of assembly-based methods were comparable to each other (**Figure 7**).

This analysis showed that compared to current genomic differences identification methods SyRI performs better. By using the structural information present in chromosome-level genome assemblies, SyRI can accurately identify breakpoints in the reference as well as the query genome for all rearranged regions.

2.3 Performance evaluation using real genomes

I compared SyRI against different methods by finding the genomic differences in the human NA19240 genome for which a chromosome-level assembly of a haplotype was recently generated (Methods: 3.5)[37]. Further, I compared these predictions against the gold-standard variation dataset that was generated by combining genomic differences

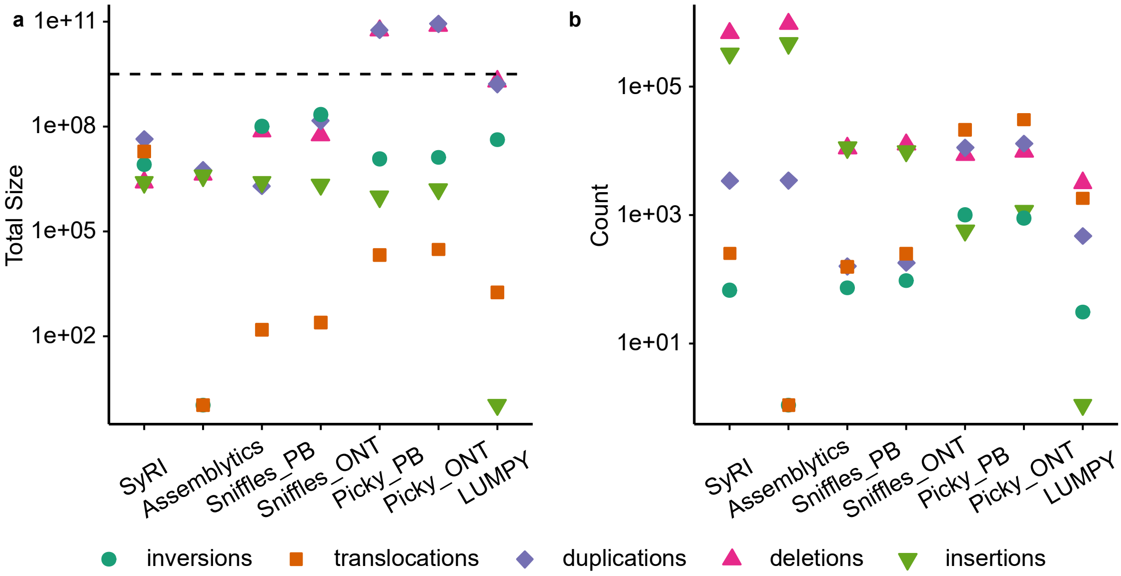


Figure 8: Total size and number of rearrangements and indels identified by different methods in the NA19240 genome. Genomic differences were identified against the human reference genome. a) The total size of annotated differences. b) The number of annotations in. The dashed line corresponds to the total size of the human reference genome assembly. Smartie-sv resulted in no output when run using real genome assemblies and hence, corresponding results are unavailable. AsmVar was computationally challenging for the human genome and therefore, it could not be analysed. For translocations, read based methods reported only single breakpoint, so size information was not available. For this figure, I considered each breakpoint to correspond to a translocation of size 1bp. (from Goel *et al.*, 2019)

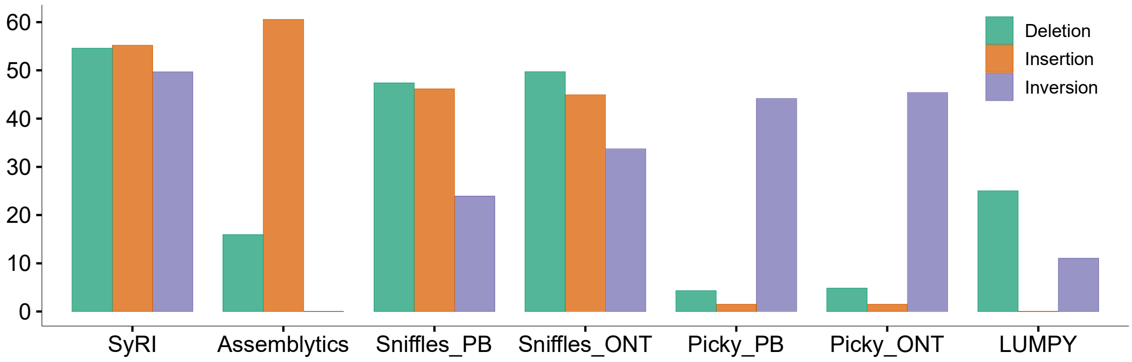


Figure 9: Gold-standard genomic differences identification by methods. Y-axis shows the per cent of gold-standard differences identified. Smartie-sv gave no output when run using real genome assemblies and hence, corresponding results are unavailable. AsmVar was computationally too challenging for the human genome and therefore, it could not be analysed. (from Goel *et al.*, 2019)

identified by various sequencing and experimental methodologies[38]. The dataset contained inversions, insertions, and deletions present in both haplotypes of the NA19240 genome compared to the human reference genome. However, this dataset was based on an older version of the human reference genome (GRCh38), so I remapped the variants to the newer version of the human reference genome (GRCh38.12) using the Genome Remapping Service from NCBI. I identified differences in the NA19240 genome using SyRI and other methods (described in the previous section, **Figure 8**) and compared the predictions against the gold-standard dataset (**Figure 9**).

SyRI identified 55.2% (9685/17545) insertions, 54.5% (9494/17391) deletions, and 49.7% (81/163) inversions from the gold standard variation data (**Figure 9**). These results were consistent with expectations as the genome assembly used to identify variants consisted of single haplotype, compared to gold-standard variation dataset that consisted of variation in both haplotypes. In this analysis too, SyRI performed better than other methods (**Figure 9**). These methods had a varying performance for the three variant types. Specifically, sniffles performed better in predicting insertions and deletions while picky performed better for inversions. Performance of Smartie-sv and AsmVar could not be analysed because Smartie-sv did not output anything whereas running AsmVar was computationally challenging for the human genome.

I also compared the methods by finding genomic differences between two accessions of *A. thaliana* (Methods: 3.5). Similar to the human genome analysis, I compared the chromosome-level assembly of accession Landsberg *erecta* (L*er*) against the reference genome sequence Col-0 (**Figure 10**)[39,40]. Assembly-based methods identified more indels compared to read-based methods. In contrast, for read-based methods the predicted total length of duplications and deletions has high. In fact, for all reads-based methods, this value was larger than the total size of the *A. thaliana* genome as they predicted multiple large deletions and tandem duplications (**Table S2, Figure 10**). These unexpected results

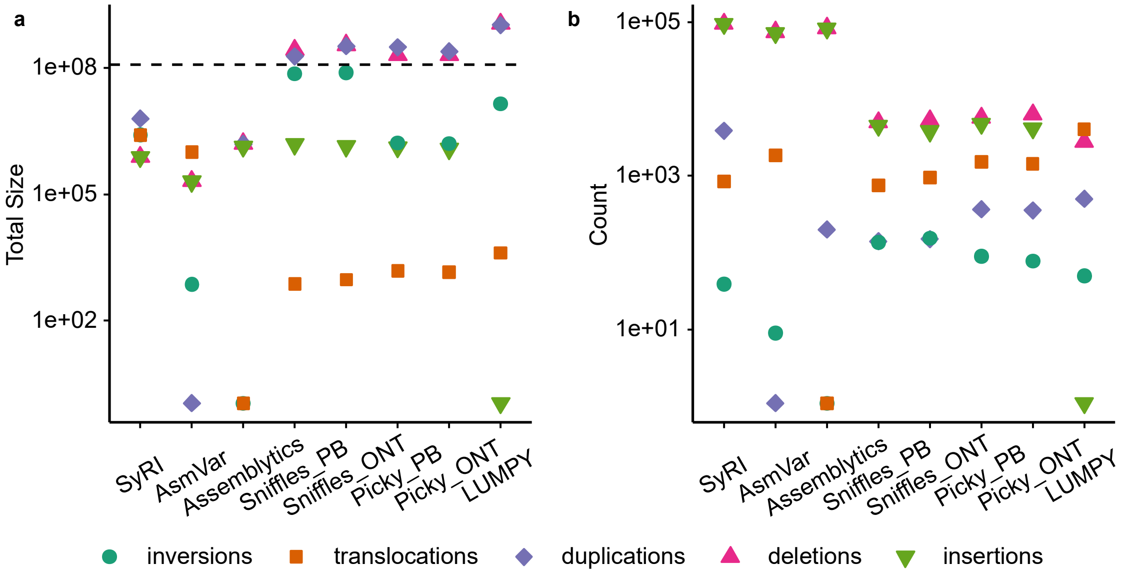


Figure 10: Total size and number of rearrangements and indels identified by different methods in the L*er* genome. Genomic differences were identified against the Col-0 reference genome. a) The total size of annotated differences. b) The number of annotations in. The dashed line corresponds to the total size of the Col-0 reference genome assembly. Smartie-sv resulted in no output when run using real genome assemblies and hence, corresponding results are unavailable. For translocations, read based methods reported only single breakpoint, so size information was not available. For this figure, I considered each breakpoint to correspond to a translocation of size 1bp. (from Goel *et al.*, 2019)

suggest false annotations of reads originating from transpositions and distal duplications as deletions and tandem duplications (**Figure 6**).

This analysis demonstrated the usefulness of whole-genome assembly-based methods for accurate identification of genomic differences.

2.4 Effect of assembly contiguity on SyRI

SyRI requires chromosome-level assemblies for accurate identification of genomic differences. This limits its applicability, as generating high-quality assemblies is still not trivial. However, this limitation can be overcome using methods like ntJoin, RaGOO etc that can generate pseudo-chromosomes from incomplete assemblies using homology against the reference genome sequence[41,42]. For cases when a reference genome is not available, I developed a heuristic method called *chroder* (for chromosome ordering) for generating pseudo-chromosomes by analyzing homology between the two incomplete assemblies (Methods: 3.6).

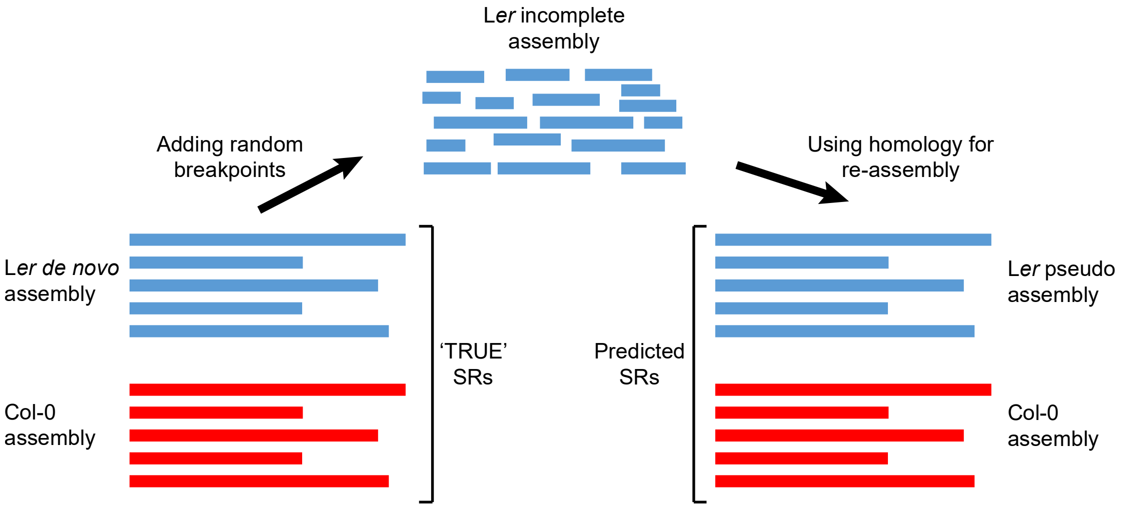

Ler incomplete assembly
Adding random breakpoints
Using homology for re-assembly
Ler de novo assembly
Col-0 assembly
'TRUE' SRs
Predicted SRs
Ler pseudo assembly
Col-0 assembly

Figure 11 Workflow for analyzing the effects of genome contiguity. Starting from the L*er* chromosome-level *de novo* assembly, an incomplete assembly was generated by adding random breaks in it. The incomplete assembly was then re-assembled based on the homology with the Col-0 reference genome. Structural rearrangements (SRs) predicted from pseudo-assemblies were compared against the rearrangements identified from the L*er de novo* assembly, which were considered as 'TRUE' SRs.

I studied the effects of assembly contiguity on SyRI's performance by performing a simulation analysis (**Figure 11**). I introduced random breaks in the chromosome-level assembly of L*er* to simulate multiple incomplete assemblies (Methods: 3.7). Then, I generated homology-based pseudo chromosomes using Col-0 as reference. Structural rearrangements between Col-0 and pseudo-genome assemblies were identified using SyRI and compared with the structural rearrangements between the reference and the chromosome-level assembly of L*er* that were considered as 'TRUE' variation data (Method: 3.7).

In this analysis, SyRI identified structural rearrangements with a sensitivity >0.9 for 90% of the pseudo-genomes with incomplete assembly N50 >470Kb. Similarly, a precision value of >0.9 was observed for 90% of pseudo-genomes with incomplete assembly N50 >674Kb (**Figure 12**). Additionally, I analysed SyRI's performance when both assemblies were incomplete by simulating incomplete assemblies from Col-0 and L*er* genomes (Methods: 3.7). I used chroder to generate homology-based pseudo-genomes and identified structural rearrangements using SyRI. Similar to earlier, structural rearrangements predicted from pseudo-genomes were compared to the true-variation data from the chromosome-level assembly. SyRI had sensitivity and precision values of

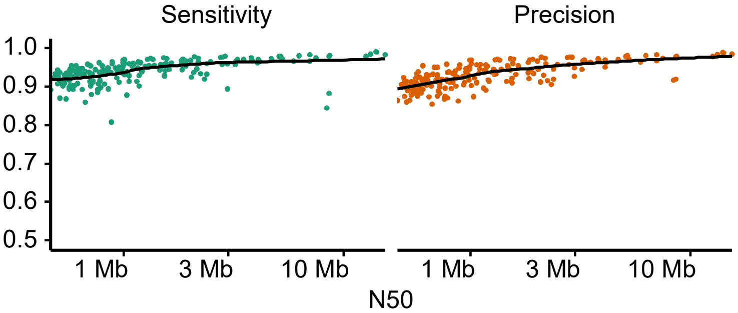


Figure 12: Rearrangement prediction efficiency of SyRI in incomplete assemblies. Each point corresponds to an incomplete assembly and the black lines represent the polynomial-fit. (from Goel *et al.*, 2019)

more than 0.7 for 70% of the pseudo-genomes with incomplete assembly N50 >868Kb and >721Kb respectively (**Figure S1**).

In this analysis, SyRI could identify almost all structural rearrangements if a chromosome-level reference genome is available. SyRI had a lower prediction quality when both assemblies were incomplete; however, it still identified many of the putative rearrangements.

2.5 Analysing multiple model species using SyRI

I identified genomic differences in humans, maize, fruit fly, and yeast using SyRI (**Table S1**, Methods: 3.8). For humans, I compared genomes NA19240 and NA12878 against the reference genome GRCh38.p12[37,43]. For maize, I compared the accession PH207 against the reference genome from B73[44,45]. For fruit fly, the genome of strain A4 was compared against the reference genome from strain ISO-1[46,47]. For yeast, the *de novo* genome assembly of strain YJM1447 was compared against the reference genome from strain S288C[48,49]. As maize is a highly repetitive genome, I masked the repeat regions to limit computational requirements[50]. In our analysis, I observed that for all organisms at least 5% of the genome was non-syntenic (**Table 1, Figure S2, Figure S3, Figure S4, Figure S5, Figure S6**).

For all these comparisons, SyRI's was computationally fast and resource-efficient. For the human, fruit fly, and yeast genomes, it required less than 600 seconds of CPU runtime and less than 1GB of memory. An exception was the maize genome for which it utilized ~3350 secs of CPU runtime and ~6GB of memory. SyRI identifies the best combination of alignments to annotate rearranged regions. In repetitive genomes, there could be many alignments between repeat regions increasing the runtime and memory requirement significantly. This problem can be alleviated by decreasing the sensitivity of whole-genome alignment and filtering out of smaller alignments.

Table 1: Computation resources used and structural differences identified by SyRI for difference genomes.

Species	Sample	CPU runtime (in secs)	Memory Usage (in MB)		Syntenic Regions	Structural Rearrangements			Un-aligned
						Inversion	Translocation	Duplication	
Human	NA12878	542.71	581	size	2.8 Gb	7.0 Mb	11.6 Mb	27.9 Mb	224.1 Mb
				% genome	91.1	0.2	0.4	0.9	7.4
				number	1147	66	270	3766	840
	NA19240	528.79	1003	size	2.8 Gb	3.7 Mb	11.8 Mb	27.1 Mb	208.8 Mb
				% genome	91.7	0.1	0.4	0.9	6.9
				number	1134	68	254	3429	848
Yeast	YJM1447	34.51	5	size	11.2 Mb	1.8 Kb	92.0 Kb	629.6 Kb	87.3 Kb
				% genome	92.5	0.02	0.8	6.0	0.7
				number	222	3	54	370	164
Fruit Fly	A4	522.02	289	size	124.8 Mb	119.5 Kb	2.0 Mb	7.5 Mb	1.2 Mb
				% genome	92.1	0.1	1.4	5.5	0.8
				number	1947	15	636	4387	1365
Maize	PH207	3342.62	5873	size	1.3 Gb	82.5 Mb	10.1 Mb	15.9 Mb	669.6 Mb
				% genome	62.2	4.0	0.5	0.8	32.5
				number	8779	195	3954	9612	15166

(from Goel et al., 2019)

2.6 Example of a highly rearranged region

Wijnker *et al.* reported two consecutive large inversions between chromosome 3 of Col-0 and L*er* accessions of *A. thaliana*[33]. In my analysis of these accessions using SyRI, I indeed identified these inversions (**Figure S7**). However, with SyRI it was also possible to identify one large translocation and one large duplication in the same region, thus further increasing the complexity of this highly rearranged region (**Figure 13**). This rearranged region overlaps multiple genes and could affect their expression. This example

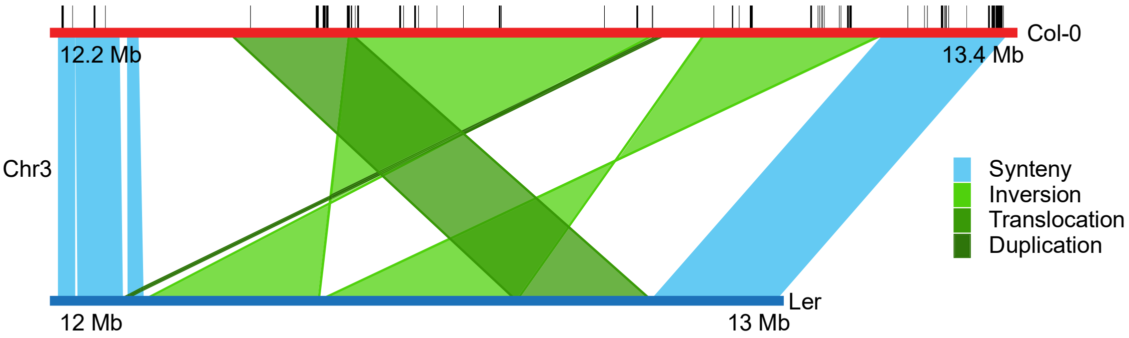


Figure 13: Multiple co-occurring rearrangements. The red and blue line reflects Col-0 and L*er* chromosome 3, respectively. Black lines on top represent regions containing genes.

illustrates the usefulness of SyRI in identifying genomic differences altering biological process.

2.7 Experimental validation of predicted translocations

Recombination can result in different copy-numbers for a translocated region in daughter cells (**Figure 14a**). This distinguishes translocations from syntenic regions, as for the later copy-number will not be affected by recombination. This phenomenon allows validation of translocations predicted by SyRI.

I used a previously published population generated by selfing F1 hybrids from crossing Col-0 and L*er* accessions of *A. thaliana*, resulting in 50 F2 recombinant plants[51]. The authors also performed whole-genome sequencing (~5x coverage/sample) and identified genotype information for each of the 50 F2 plants by aligning the sequencing-reads to Col-0 reference genome and using TIGER for crossover identification[52]. I selected all translocations (and transpositions) that were larger than 1kb and outside the peri-centromeric regions (n=117) and estimated their expected copy-numbers in all samples based on the genotype of that sample. As reads originating from both loci of a translocated region would align to the same loci in the reference sequence, by analyzing the read-coverage at the translocated region, I calculated the observed copy-number for the

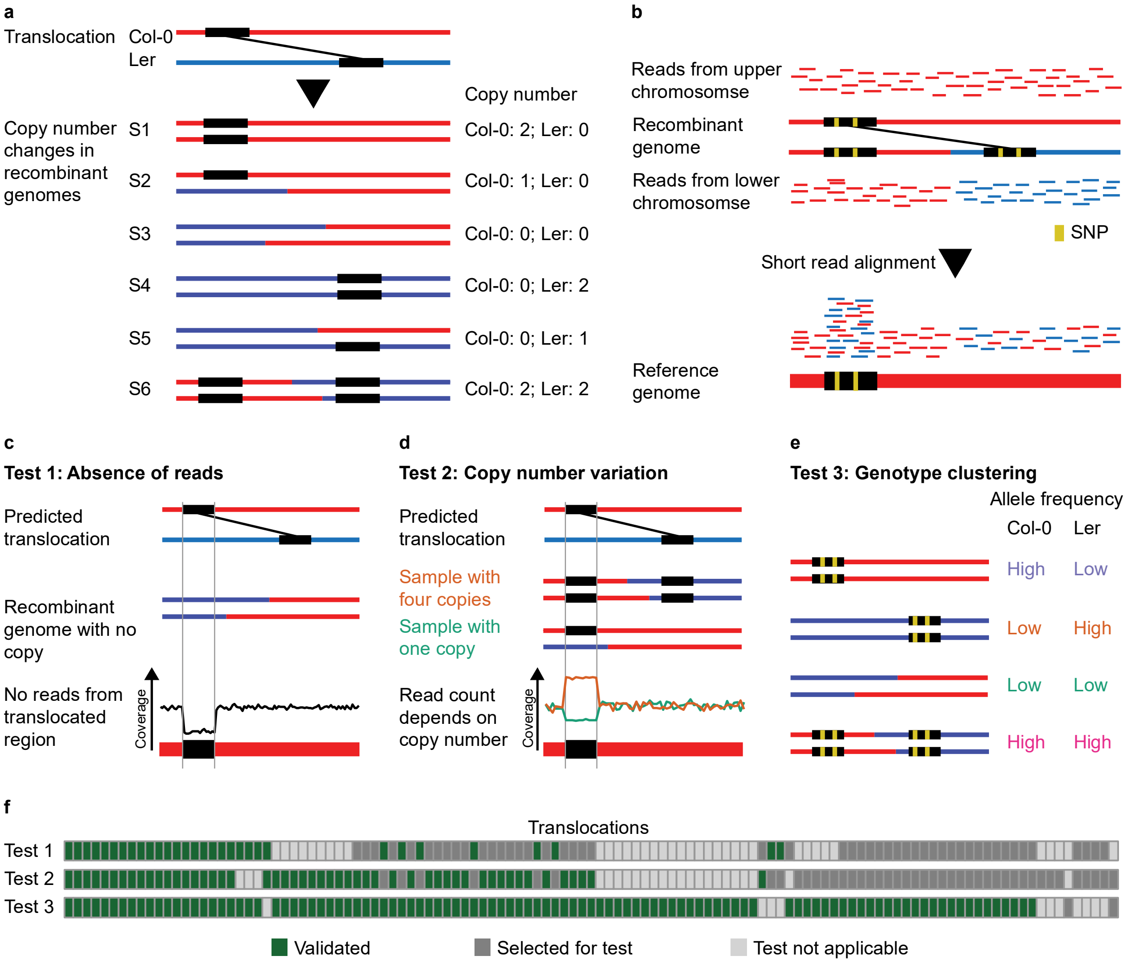

a
Translocation
Col-0
Ler
Copy number
Copy number changes in recombinant genomes
S1
Col-0: 2; Ler: 0
S2
Col-0: 1; Ler: 0
S3
Col-0: 0; Ler: 0
S4
Col-0: 0; Ler: 2
S5
Col-0: 0; Ler: 1
S6
Col-0: 2; Ler: 2
b
Reads from upper chromosomse
Recombinant genome
Reads from lower chromosomse
SNP
Short read alignment
Reference genome
c
Test 1: Absence of reads
Predicted translocation
Recombinant genome with no copy
No reads from translocated region
Coverage
d
Test 2: Copy number variation
Predicted translocation
Sample with four copies
Sample with one copy
Read count depends on copy number
Coverage
e
Test 3: Genotype clustering
Allele frequency
Col-0
Ler
High
Low
Low
High
Low
Low
High
High
f
Translocations
Test 1
Test 2
Test 3
Validated
Selected for test
Test not applicable

Figure 14: Validating translocations using recombination induced copy-number changes. (a) Recombination between translocated loci can result in copy-number differences in recombinant genomes. (b) Short-reads from recombinant genomes are aligned to the reference genome. (c-e) Tests used for validation of the predicted translocations: (c) testing for the absence of reads in samples, (d) goodness-of-fit between expected and observed copy-number, and (e) clustering of samples having the same genotypes. (f) In the heatmap, rows correspond to the tests and columns represent individual translocations. The colour represents whether a translocation was validated (green), was selected but could not be validated (dark grey), or was filtered out as the test was not applicable (grey). (from Goel *et al.*, 2019)

translocation (**Figure 14b**). Additionally, read counts for reference and alternate alleles in the translocated region were also identified.

Predicted translocations were tested using three tests and I considered a translocation validated if it passed any of the three tests (**Figure 14c-e**). In the first test, I checked samples with genotypes corresponding to the absence of a translocated region and considered a translocation as valid if these samples had no copy (read coverage <0.2x) of the translocated region (Method: 3.9.1). In the second test, I assessed the linear model fit between the expected and observed copy-numbers of a translocated region across samples and considered translocations with good fitting as valid (Method: 3.9.2). In the third test, I compared the read counts of Col-0/L*er* alleles in samples with the same genotypes and considered translocations for which read counts clustered based on genotypes as valid (Method: 3.9.3). More than 90% (108/117) of the predicted translocations were validated by at least one test and 50% (59/117) translocations with at least two tests (**Figure 14f**). I checked the read alignments for the remaining translocations manually and though I could observe signals supporting their existence, these signals were not strong enough to be validated by these tests. From this analysis, I conclude that SyRI accurately identified genome-wide translocations.

3 Methods

Parts of this chapter were the basis of a manuscript that was published as a peer-reviewed research article in Genome Biology (Goel et al. 2019)[1]*. However, in this chapter, I only discuss the work that was done by me. Results and data that were not generated by me are either clearly pointed or cited.*

Authors list (Goel et al. 2019): Manish Goel[α]*, Hequan Sun*[β]*, Wen-Biao Jiao*[α]*, Korbinian Schneeberger*[α, β]*.*

Author affiliations: [α]*Department of Chromosome Biology, Max Planck Institute for Plant Breeding Research, 50829 Cologne, Germany;* [β]*LMU Munich, 82152 Planegg-Martinsried, Germany.*

Authors contributions (Goel et al. 2019): The project was conceived by KS and WBJ. MG and KS developed the algorithms. MG implemented SyRI and performed all analyses. HS processed recombinant genome sequencing data and identified crossing-over sites. WBJ generated the Ler assembly. The manuscript was written by MG and KS with inputs from HS and WBJ. All authors read and approved the final manuscript.

In this section, I provide additional description of SyRI's workflow and algorithms. Additionally, I also describe the steps and strategies used in performing all the analysis.

3.1 Syntenic region identification

Starting from whole-genome alignments, SyRI selects all directed alignments between a pair of homologous chromosomes and generates a directed acyclic graph (DAG). In this DAG, each node corresponds to an alignment and edges are added between two nodes when the corresponding aligned regions are:

- collinear in both genomes (i.e. have the same relative position in both genomes) irrespective of the distance between them
- not overlapping
- not separated by other collinear alignments on both genomes

In **Figure S8**, I show an example of this process. Starting from alignments *a-l*, SyRI generated a DAG with each alignment becoming a node. Edges were added from 'a' to 'e' as well as 'c' as both nodes were collinear to 'a', but no edge was added from 'd' to 'e' as their aligned regions were not collinear. Nodes 'a' and 'b' were not connected as the aligned regions were overlapping, whereas nodes 'a' and 'd' were not connected as they were separated by 'c' on both genomes.

SyRI assigns each node a score corresponding to its alignment score. It also adds two imaginary 0-score nodes, S (start) and E (end) and connects node S to all nodes without any in-edge and node E to all nodes without any out-edge. It identifies all alignments that constitute the syntenic region by finding the longest path from node S to E using dynamic programming. For all pair of homologous chromosomes, SyRI repeats this process.

3.2 Inversion identification

Both synteny and inversion are characterized as having a conserved location in the two genomes. This implies that reverse complementing the query genome would transform alignments of an inversion into locally syntenic regions. Additionally, alignments representing a putative inversion can have different conformations (**Figure S9**). Overlapping inverted alignments result in conflicting annotations further complicating inversion identification. In SyRI, I developed a methodology that selects inverted alignments corresponding to the longest inversions for all conformations of inverted regions.

For inversion identification, SyRI selects all inverted alignments between a pair of homologous chromosomes and then reverse complements the query chromosome converting the inverted alignments to directed alignments (**Figure S10**). Similar to syntenic region identification, it creates a DAG where each node corresponds to a now directed alignment and edges are added between two nodes when the corresponding aligned regions are:

- collinear in both genomes (i.e. have the same relative position in both genomes) irrespective of the distance between the regions
- not overlapping
- not separated by other collinear alignments on both genomes

SyRI uses alignment scores as the node score and adds two imaginary 0-score nodes, Start (S) and End (E). Edges are added from node S to all nodes and from all nodes to node E. In this DAG, each path from S to E represents a putative inversion candidate with its score being the difference of the sum of the score of its constituent alignments' and the sum of alignment score of any syntenic region within the inverted region. This is because syntenic regions cannot be present within an inversion and therefore would be removed if any overlapping candidate inversion were selected.

The candidates can overlap each other and could have conflicting annotations. SyRI selects all non-conflicting inversions with highest total alignment score. For this, it generates a DAG with candidates as nodes and edges between nodes representing candidates that do not overlap, are collinear, and are not separated by another candidate. Candidate score is used as the node score. Two imaginary 0-score Start (S) and End (E) nodes are added with edges from node S to all nodes without any in-edge and to node E from all nodes without any out-edge. SyRI finds non-conflicting candidates by finding the longest path from node S to node E using dynamic programming.

3.3 Translocation and duplication (TD) identification

From the remaining alignments, SyRI selects translocations and duplications (TDs) and removes redundant (repetitive) alignments. Genomic repeats result in multiple overlapping alignments with conflicting annotations. SyRI accurately identifies TDs from them using a two-step methodology. First, it identifies candidate TDs that consists of one or more alignments representing a putative relocation event (**Figure S11a**). Second, SyRI selects non-conflicting candidates with the highest alignment score (**Figure S11b**).

3.3.1 Overlapping candidates have inter-dependent annotations

Overlapping candidates can influence the annotation of a candidate TD. I explain this in **Figure S11c** where four candidates (two green and two blue) overlap each other. In the first case, the green candidates are longer and better represent two relocation event; therefore, they should be selected as TDs. In the second case, however, the blue candidates are longer and better represent TDs suggesting that the green candidates are redundant. Even though green candidates are the same in both cases, their annotation is different because of overlapping candidates. Therefore, SyRI compares all overlapping candidates simultaneously to find optimal annotations for rearranged regions.

3.3.2 Candidate TD identification

SyRI generates a DAG with the remaining directed alignments, where each node corresponds to an alignment, and adds edges between nodes when the corresponding aligned regions are:

- collinear in both genomes (i.e. have the same relative position in both genomes) irrespective of the distance between the regions
- not overlapping
- not separated by other collinear alignments on both genomes
- not separated by a syntenic or an inverted region on both genomes

The alignment score is used as the node score. SyRI adds two imaginary 0-score Start (S) and End (E) nodes with edges from node S to all other nodes and to node E from all other nodes (**Figure S12**). In this graph, each path from S to E corresponds to a candidate TD. SyRI finds them using dynamic programming and calculates candidate score for each candidate using its alignment length and gaps between them:

$$score = \ min\left(\frac{genA\_aligned\_length - genA\_gap\_length}{genA\_aligned\_length}, \frac{genB\_aligned\_length - genB\_gap\_length}{genB\_aligned\_length}\right)$$

Candidates with large gaps get a negative score and are filtered out. A similar process is followed for inverted alignments.

3.3.3 Selecting optimal candidate TDs

SyRI groups overlapping candidate TDs generating a network of rearranged repeat regions. Starting from a candidate TD, SyRI adds candidates that overlap the focal candidate in the group. Then, it iteratively adds candidates overlapping with the recently added candidates until no new candidate can be added. In **Figure S11c**, starting from the left green candidate, SyRI would first add the two blue candidates as they overlap the green candidate. Then, it would add the right green candidate as it overlaps the blue candidates. Consequently, all four candidates would constitute a network.

From such a network, SyRI selects candidates aligning regions that do not overlap with other candidates or syntenic/inverted regions (**Figure S13a**). Since only one candidate can annotate such a region, therefore, SyRI selects such candidates as *necessary*. It removes *redundant* candidates that overlap with syntenic/inverted regions or already selected candidates on both genomes (**Figure S13a**). SyRI repeats this process until it reaches a *deadlock* when it cannot select or remove any more candidates (**Figure S13b**). To overcome deadlocks, it uses brute-force (for networks with <50 candidates) and randomized-greedy (networks with >50 candidates) methods.

In the brute-force method, SyRI selects all different combinations of non-conflicting candidates. SyRI selects necessary candidates as the initial combination. Then, it iteratively checks whether a candidate is non-conflicting to any combination. If yes, then it creates a new combination comprising of candidates of the non-conflicting combination and the focal candidate (**Figure S13c**). After iterating over all candidates, SyRI compares the score for each combination and selects the highest scoring combination (see below). If the number of combinations becomes large, then it switches to the randomized-greedy method to restrict computational resources usage. In the randomized-greedy method, SyRI again selects necessary candidates as output combination. To overcome a deadlock, it randomly adds one of the twenty highest-scoring candidates to the output combination with selection probability for a candidate proportional to its candidate score. SyRI then continues to find necessary/redundant candidates while resolving deadlocks with random selection until all candidates are either selected or removed resulting in a combination of non-conflicting candidates. SyRI repeats this process 100 times to get 100 combinations of non-conflicting candidates and selects the highest scoring combination.

I defined combination score as the number of unannotated bases its constituent candidates annotates. SyRI selects a combination with a high score and a low number of candidates. For combinations with similar scores, the combination comprising of few longer candidates would be preferred over combinations comprising of many smaller candidates. SyRI annotates candidates of the selected combination as translocations or duplications based on their overlap with syntenic/inverted regions and with other candidates. A candidate is annotated as translocation when it does not overlap with syntenic/inverted regions and other candidates on both genomes, and duplication when it overlaps with syntenic/inverted regions on a genome. If two candidates overlap each other, then the candidate with the higher score is selected as translocation and the other is selected as duplication.

3.4 Simulation analysis

3.4.1 Generating rearranged genomes from the *A. thaliana* reference genome

I used SURVIVOR and RSVSim to simulate 100 rearranged genomes each with inversions, transpositions, translocations, tandem duplications, and distal duplications from the Col-0 assembly[53,54]. I simulated 40, 436, 100, 100, and 1241 rearrangements, respectively, for the five rearrangement types. I used the Col-0 vs L*er* comparison to get a size distribution for inversions, transpositions, and distal duplications, whereas for translocations and tandem duplications size ranged from 1000-5000bp and 100-1000bp respectively. Using SURVIVOR, I also simulated 100 genomes each having 1000 indels with size ranging from 1-500bp.

3.4.2 Simulating reads from the rearranged genomes

For read-based genomic differences identification methods, I simulated reads from the rearranged genomes using wgsim (Illumina short-reads, parameters: -e 0.001 -d 550 -N 12000000 -1 150 -2 150) and SURVIVOR (PacBio and Nanopore reads, default parameters) to get 30x genome coverage[53,55]. SURVIVOR required read-profiles for simulating reads which I generated using the long-read sequencing data from NCBI project: PRJEB21270[56]. I aligned the reads to Col-0 assembly using minimap2 and converted the alignments from SAM to BAM format using samtools[57,58].

3.4.3 Finding genomic differences using various methods

SyRI

Rearranged genome assemblies were aligned using nucmer (parameters: --maxmatch -c 100 --noextend -l 100). Alignments were filtered using delta-filter (parameters: -m –i 90 -l 100) and converted to tab-separated table format using show-coords (parameters: -THrd). SyRI was run using the default parameters.

Assemblytics[28]

I used the same alignments as I used for SyRI. The default value for unique sequence length was used with variant size set to 1-100000 bp.

AsmVar[27]

I followed the pipeline provided by the authors of the method. Reference genome was indexed using lastdb with default parameters. I aligned genomes using lastal and then used last-split with parameters suggested by the authors. Finally, I used ASV_VariantDetector with default parameters to predict genomic differences[59].

Smartie-sv[29]

I used the method with default settings but changed the number of jobs to run in parallel and job wait time to make it suitable for available computational resource.

Sniffles[30]

Simulated long-reads (both PacBio and Nanopore) were aligned using minimap2[57]. Using samtools, the SAM file was converted to BAM, and subsequently sorted and indexed[58]. Sniffles was run using the default parameters.

Picky[31]

I used the workflow as described by the authors. Simulated reads were converted from fasta to fastq format using faToFastq[60]. I used lastal to align reads and identified structural differences with picky. I used suggested parameters for all commands.

LUMPY[32]

I aligned short-reads using minimap2[57]. The alignments were processed using samblaster and samtools as suggested by authors[58,61]. I used LUMPY with default parameters but changed the values of paired-end read distribution parameters (mean: 550, read_length: 150, min_non_overlap: 150) to match simulated reads.

3.4.4 Calculating prediction accuracy

I calculated the efficiency of different methods by comparing the coordinates of the breakpoints of simulated and predicted rearrangements. I considered the predicted breakpoints for inversions, transpositions, translocations, tandem duplications, and distal duplications correct when they were within 150bp range of simulated breakpoints. Read-based methods did not annotate transpositions or translocations but reported breakpoints corresponding to translocations and transpositions. Consequently, I used breakpoints as representatives for these rearrangements. Similarly, as tandem and distal duplications were not distinguished, all annotations corresponding to duplicated regions were considered. For indels, I checked whether a predicted indel is within N bps of any simulated indel with its size also within N bps to simulated indel's size. I used two different values for N: 5 bps and 100 bps.

3.5 Comparing real genomes and comparison against the gold-standards dataset

I aligned the assemblies of NA19240 and L*er* genomes to the respective human and *A. thaliana* reference genome as described in section 3.8 and simulated reads as described in section 3.4.2. For simulating long reads for NA19240, I used the read-profile provided by SURVIVOR[53]. I identified structural differences using different methods as described in section 3.4.3. However, Smartie-sv did not result in any output for both assemblies and AsmVar was computationally too challenging for NA19240, therefore their results are not available. AsmVar had better predictions in the .svd output file compared to the .vcf output file; therefore, I used it for downstream analysis. I used custom scripts to extract count and length of the predicted insertions, deletions, and inversions in somatic chromosomes. I considered an insertion (or deletion) as correctly identified if a method predicts an insertion (or deletion) within 100 bp region from it. I considered an inversion correctly identified if a method predicted an overlapping inversion. As read-based used to generate gold-standard data might have misrepresented inverted TDs as inversions, for SyRI, I checked overlap with them as well.

3.6 Generating homology-based pseudo-chromosome level assemblies

I developed a heuristic method called 'chroder' (for chromosome ordering) to generate homology-based pseudo-chromosomes using whole-genome alignments between incomplete assemblies. Chroder can use homology between a chromosome-level and an incomplete assembly as well as between two incomplete assemblies to generate pseudo-chromosomes. In the presence of chromosomes, chroder uses them as a template to order and orient contigs from the incomplete assembly. When both assemblies are incomplete, chroder uses alignments between homologous contigs to order and orient them. If a contig aligns with multiple contigs from the other assembly, then chroder orders these contigs to form a longer sequence (**Figure S14**). Chroder orients homologous contigs to have directed alignments by reverse complementing the contigs with inverted alignments; however, it does not break contigs and assigns a contig to only one pseudo-chromosome. Finally, it concatenates contigs with Ns in between them to generate pseudo-chromosomes. Contigs that could not be assigned to pseudo-chromosomes are filtered out.

3.7 Incomplete assemblies generation and analysis

3.7.1 Single incomplete assembly

I simulated incomplete assemblies from the L*er* genome by introducing 10-400 random breaks to generate 200 contig assemblies. Using RaGOO, I generated homology-based pseudo-chromosome level assemblies with Col-0 as a reference, which were then aligned to Col-0 using nucmer. I considered genomic differences identified by SyRI from pseudo-genome assemblies as correct if a rearrangement of the same type existed in the Col-0 vs L*er* comparison within 100 bps.

3.7.2 Double incomplete assemblies

I simulated incomplete assemblies from Col-0 and L*er* genomes by introducing 10-400 random breaks to generate 100 pairs of incomplete assemblies. I used chroder to generate homology-based pseudo-chromosome level assemblies. This failed for sixteen pairs so I filtered them out from the further analysis. I aligned these assemblies using nucmer, identified genomic differences using SyRI, and calculated prediction correctness based on the Col-0 and L*er* assembly predictions.

3.8 Aligning whole-genome assemblies

I used eleven whole-genome assemblies from five model species, three from humans and two each form *A. thaliana*, yeast, fruit fly, and maize (**Table S1**). I selected only chromosomes and filtered-out unplaced scaffolds/contigs. I aligned the assemblies using nucmer from the MUMmer3 package[36]. For each species, I selected nucmer parameters -c, -b, and -l to ensure sufficient alignment resolution while limiting computational runtime and resource requirements. Also, I used --maxmatch parameter to get all alignments. I selected alignments with identity >90% and length >100bp using delta-filter and transferred alignments to a tab-separated table format using show-coords[36]. The exact commands used are shown in **Table S3**. The highly repetitive maize genomes were repeat-masked using RepeatMasker v4.0.6[62].

3.9 Validation of predicted translocations

I extracted allele count information from the whole-genome sequencing data of 50 F_2 recombinants using SHORE[63]. For calculating copy-number of a predicted translocation in a sample, I divided the average read-coverage for the translocated region by the average read-coverage for the entire sample. I filtered out translocations in the peri-centromeric regions and for which >25% of the sequence had >10% Ns. Within the translocated regions, I selected highly conserved SNPs and filtered out SNPs having sequence variation within 25bps to it.

3.9.1 Test 1: Absence of translocated sequence

I tested translocations for which genotype of at least two samples indicated the absence of the translocated region. I considered a predicted translocation as valid if all samples with the predicted absence of region had an average read-coverage of less than 0.2x in the translocated region.

3.9.2 Test 2: Modelling expected and observed copy numbers

For a translocation, I filtered out samples that had either homozygous Col-0, heterozygous, or homozygous L*er* genotype at both loci of the translocation. In these three cases, the translocated region has two copies each increasing the number of samples with two expected copies biasing linear modelling. I tested only those translocations that had samples corresponding to at least three different expected copy-number values. I used the lm function of R to generate a linear regression model and did multiple hypothesis correction using Benjamini-Hochberg method[64]. I considered a translocation valid if the model had slope >0.75 and p-value $<10^{-6}$.

3.9.3 Test 3: Genotype derived sample clustering

For SNP markers in the translocated regions, I normalized allele-counts for each sample based on the number of reads sequenced and the number of reads mapped. I filtered out outlier markers (highest one percentile allele count). Translocations with at least:

- three SNP marker positions
- two genotypes with three samples each

were tested. For a translocation, I filtered out genotypes with less than three samples. Translocations with low alternate allele count variance (variance<1) were also removed.

For a translocation, I represented each sample as a point on the 'reference allele count vs alternate allele count' plane and calculated Euclidean distance between points. To quantify clustering of samples corresponding to a genotype, I calculated *closeness_score* as the sum of ratios of the mean distance of samples of a genotype to the mean distance against samples from other genotypes. For true translocations, samples from the same genotype would have similar allele counts (small distance) compared to samples from different genotypes with different allele clusters (higher distance) resulting in lower closeness_score. I created a null distribution of closeness_score by simulating clusters of genotypes where allele counts (for reference and alternate alleles) were sampled from Poisson distribution. For each translocation, I calculated lower-tail p-value by comparing its closeness_score against the null distribution. I did multiple hypothesis correction using the Benjamini-Hochberg method and selected translocations with p-value<0.05 as valid[64].

4 Discussion and Conclusion

Parts of this chapter were the basis of a manuscript that was published as a peer-reviewed research article in Genome Biology (Goel et al. 2019)[1]*. However, in this chapter, I only discuss the work that was done by me. Results and data that were not generated by me are either clearly pointed or cited.*

Authors list (Goel et al. 2019): Manish Goel[α]*, Hequan Sun*[β]*, Wen-Biao Jiao*[α]*, Korbinian Schneeberger*[α, β]*.*

Author affiliations: [α]*Department of Chromosome Biology, Max Planck Institute for Plant Breeding Research, 50829 Cologne, Germany;* [β]*LMU Munich, 82152 Planegg-Martinsried, Germany.*

Authors contributions (Goel et al. 2019): The project was conceived by KS and WBJ. MG and KS developed the algorithms. MG implemented SyRI and performed all analyses. HS processed recombinant genome sequencing data and identified crossing-over sites. WBJ generated the Ler assembly. The manuscript was written by MG and KS with inputs from HS and WBJ. All authors read and approved the final manuscript.

4.1 Whole-genome assembly based genomic differences identification

Accurate identification of genomic differences is critical for understanding biological diversity. During my PhD, I developed a novel computational method for identifying all genomic differences between two closely related organisms by comparing their chromosome-level assemblies. I call this method SyRI and it identifies syntenic (structurally conserved) as well as structurally rearranged regions that have a different location, orientation, and/or copy-number between genomes.

4.1.1 Changing the paradigm of structural rearrangement

Single nucleotide polymorphisms (SNPs) are the most commonly studied genomic variation. They are identified by first aligning sequencing-reads to a reference genome and then finding positions with multiple reads with a mismatched base. Multiple structural rearrangement identification methods have been developed using a similar strategy. These methods sequentially analyse the alignment breakpoints between sequencing reads or contigs (from the query genome) and the reference genome to identify structural rearrangements. However, this approach does not consider the overlap and conflicts between annotations resulting from repeat alignments. Additionally, as read-based methods do not have query genome assembly, they cannot annotate rearrangement coordinates accurately in the query genome.

SyRI overcomes these challenges. It analyses sequence as well as structure information of the query genome, using whole-genome alignments between chromosome-level assemblies, to identify genomic differences with high accuracy. SyRI’s novel approach allows comparison of overlapping alignments for identification of non-conflicting rearrangements. This is a significant technical advancement compared to current methods as now genome-wide optimised rearrangements can be identified.

4.1.2 Genome-wide optimisation for rearrangements

SyRI uses a novel approach for structural rearrangement identification where it first identifies genome-wide syntenic regions (longest set of collinear alignments). Syntenic regions describe structurally conserved regions, implying that non-syntenic regions are structural rearrangements by definition. Thus, identifying syntenic regions simultaneous identifies all rearranged regions as well. This approach transforms the challenging problem of rearrangement identification to a comparatively easier problem of rearrangement annotation.

SyRI annotates rearranged regions as inversions, translocations, and duplications based on their location, orientation, and copy-number. Using genome-graphs and optimisation algorithms, it compares overlapping alignments and performs simultaneous identification of non-conflicting rearrangements that best explain genomic structural differences. This makes SyRI, to the best of my knowledge, the first method to identify structural rearrangements optimised for genome-wide differences. Finally, even though I used the algorithms and methods mentioned here for analysing genome-graphs generated from whole-genome alignments between two genomes, they can be extended to other genome-graphs as well[65,66].

4.1.3 Nested sequence differences and their identification

SyRI also identifies sequence differences (structural variation, indels, and SNPs) in the aligned regions and between adjacent alignments of all syntenic and rearranged regions. This generates a hierarchy of genomic differences where sequence differences are present within larger structurally rearranged regions (e.g. a SNP in a translocated region). As recombination is suppressed in rearranged regions, sequence differences can accumulate[7]. Also, recombination between relocated loci of a translocation or duplication can result in copy-number variation. Further, as rearranged regions do not follow Mendelian segregation, SNPs in them can confound the variance in genotypes affecting the interpretation of genomic patterns from recombination analysis, selection screens, and genome-wide association studies[33,34]. Using SyRI it is now possible to filter out such rearranged SNPs.

4.2 Efficient genomic differences identification by SyRI

I analysed SyRI's performance in both simulated and real genomes and showed that it can identify structural rearrangements accurately in both datasets. For simulated rearranged genomes, it had near 100% sensitivity implying that it can identify all simulated rearrangements without many false positives. By analysing the human

NA19240 genome, for which gold-standard variation data was available, I also demonstrated that SyRI accurately identifies genomic differences from real-genomes as well. In comparison, current methods (both assembly and read-based) had limited performance. Assembly-based methods were limited by design as they were not developed for identification of all different classes of rearrangements (AsmVar cannot identify duplications, Assemblytics cannot find inversions and translocations, Smartie-SV cannot find translocations and duplications). Read-based methods could identify more classes of rearrangements but were unable to identify all corresponding breakpoints. Additionally, read-based methods were unable to distinguish between breakpoints from relocated regions versus local genomic differences. Consequently, breakpoints corresponding to transpositions and intra-chromosomal distal duplications were incorrectly annotated as large deletions or tandem duplications. For cases when chromosome-level assemblies are not available, I showed that SyRI can still be used with homology-based pseudo-chromosome level assemblies to gain useful insights about the structural rearrangements.

4.3 Current limitations of SyRI

SyRI analyses different combinations of alignments for TDs identification allowing identification of non-conflicting annotations from highly overlapping repeat. However, listing and analysing all TDs is computationally challenging. If a large TD is represented by multiple alignments, then SyRI generates all possible candidate TDs from these alignments which can result in 2^n candidates (n = number of alignments). This can significantly increase RAM and CPU usage. I am working to optimise the algorithms developed in this work to improve SyRI's performance in such worst cases in collaboration with Prof. Dr Gunnar W. Klau, Institute of Informatics, Heinrich Heine University, Düsseldorf.

SyRI also lacks some ease-of-life features. Currently, it requires that the same strands of the homologous chromosomes are compared. Whole-genome aligners do not consider strand information. So, if different strands are aligned then biologically syntenic regions result in inverted alignments. As SyRI cannot differentiate between inverted alignments corresponding to inversions and those originating from different strands, it will assume that the homologous chromosomes are inverted and do not have syntenic region. The current solution for this problem is for the user to ensure that the same strands are being compared. This issue also reflects that analysing genomes using SyRI is a two-step process: performing whole-genome alignment and finding genomic differences. A more user-friendly approach would involve using assemblies as input, perform whole-genome alignment and required pre-processing internally, and then identify genomic differences.

4.4 Towards population-level genomic variance identification

One of the biggest goal of genomic research at the beginning of the 21st century was the accurate generation of whole-genome assembly. With the advances in sequencing technologies and the development of ingenious assembly methodologies, this goal has been nearly achieved. This claim is supported by multiple recent studies that sequenced several organisms of a species and generated reference-quality genome assemblies[39,48,67,68]. Scientists throughout the world are constantly sequencing new organisms, creating an ever-increasing database of whole-genome assemblies. However, little progress has been made towards the population-level analysis and comparison of these assemblies. SyRI provides an excellent platform for the development of the first method for population-wide genomic differences identification by comparing multiple whole-genome assemblies. Technically, such an analysis would be similar to generating a pan-genome as it will try to find all conserved genomic regions in a population. However, it would also describe the variant regions and therefore, increase our understanding of genic presence-absence variation, differences in regulatory regions, as well as distribution of transposable elements in a population. Identification of large conserved rearrangements in a population could provide information about population stratification that can potentially lead to speciation. A multi-genome comparison would also be helpful to learn about the current evolutionary pressure within a species as well as to decipher the evolutionary history.

4.5 Concluding remarks

During my PhD research, I developed SyRI, a novel computational method for the identification of genomic differences from whole-genome assemblies. SyRI uses a novel strategy where it first identifies structurally conserved regions, and then annotates rearranged regions using graph and optimisation strategies. I compared SyRI against the current genomic differences identification methods using simulated and real genomes and demonstrated that SyRI outperforms all current methods. I also demonstrated the advantages of using whole-genome assembly compared to sequencing reads for structural rearrangement identification. I analysed genomes of five model species and showed that SyRI can efficiently identify genomic differences from genomes of all complexities. Finally, I validated more than 100 translocations predicted by SyRI genetically by using a hybrid population of two accessions of *A. thaliana*.

I believe SyRI would contribute towards initialising a new phase in genomics. It will allow a transition from the contemporary genome comparison strategies focussed mainly on SNVs and small indels to the analysis of more consequential structural rearrangements. Future technological developments would lead to even better assemblies increasing the efficiency of assembly-based methods like SyRI. Additionally, identification of nested sequence variation by SyRI will help improve the accuracy of the current marker-based analysis. Finally, the algorithms developed in this work would support the development of methods for performing population-level multiple-genome comparisons.

5 Bibliography

1. Goel, M., Sun, H., Jiao, W.-B. & Schneeberger, K. SyRI: finding genomic rearrangements and local sequence differences from whole-genome assemblies. *Genome Biol.* **20**, 277 (2019).
2. Hiyoshi, A., Miyahara, K., Kato, C. & Ohshima, Y. Does a DNA-less cellular organism exist on Earth? *Genes to Cells* **16**, 1146–1158 (2011).
3. Crick, F. Central dogma of molecular biology. *Nature* **227**, 561–563 (1970).
4. Lupski, J. R. Structural variation mutagenesis of the human genome: Impact on disease and evolution. *Environ. Mol. Mutagen.* **56**, 419–436 (2015).
5. Conrad, D. F. *et al.* Mutation spectrum revealed by breakpoint sequencing of human germline CNVs. *Nat. Genet.* **42**, 385–391 (2010).
6. Chiang, C. *et al.* The impact of structural variation on human gene expression. *Nat. Genet.* **49**, 692–699 (2017).
7. Rowan, B. A. *et al.* An Ultra High-Density Arabidopsis thaliana Crossover Map That Refines the Influences of Structural Variation and Epigenetic Features. *Genetics* **213**, 771–787 (2019).
8. Sudmant, P. H. *et al.* An integrated map of structural variation in 2,504 human genomes. *Nature* **526**, 75–81 (2015).
9. Weischenfeldt, J., Symmons, O., Spitz, F. & Korbel, J. O. Phenotypic impact of genomic structural variation: insights from and for human disease. *Nat. Rev. Genet.* **14**, 125–138 (2013).
10. Quigley, D. A. *et al.* Genomic Hallmarks and Structural Variation in Metastatic Prostate Cancer. *Cell* **174**, 758-769.e9 (2018).

11. Lupiáñez, D. G. *et al.* Disruptions of topological chromatin domains cause pathogenic rewiring of gene-enhancer interactions. *Cell* **161**, 1012–1025 (2015).

12. Malhotra, D. *et al.* High frequencies of de novo cnvs in bipolar disorder and schizophrenia. *Neuron* **72**, 951–963 (2011).

13. Xu, B. *et al.* Strong association of de novo copy number mutations with sporadic schizophrenia. *Nat. Genet.* **40**, 880–885 (2008).

14. Levy, D. *et al.* Rare De Novo and Transmitted Copy-Number Variation in Autistic Spectrum Disorders. *Neuron* **70**, 886–897 (2011).

15. Zhang, F., Gu, W., Hurles, M. E. & Lupski, J. R. Copy Number Variation in Human Health, Disease, and Evolution. *Annu. Rev. Genomics Hum. Genet.* **10**, 451–481 (2009).

16. Perry, G. H. *et al.* Diet and the evolution of human amylase gene copy number variation. *Nat. Genet.* **39**, 1256–1260 (2007).

17. Nair, S. *et al.* Adaptive Copy Number Evolution in Malaria Parasites. *PLoS Genet.* **4**, e1000243 (2008).

18. Emerson, J. J., Cardoso-Moreira, M., Borevitz, J. O. & Long, M. Natural selection shapes genome-wide patterns of copy-number polymorphism in Drosophila melanogaster. *Science* **320**, 1629–1631 (2008).

19. Friedman, A. R. & Baker, B. J. The evolution of resistance genes in multi-protein plant resistance systems. *Current Opinion in Genetics and Development* **17**, 493–499 (2007).

20. Vlad, D. *et al.* Leaf shape evolution through duplication, regulatory diversification and loss of a homeobox gene. *Science* **343**, 780–783 (2014).

21. Soyk, S. *et al.* Duplication of a domestication locus neutralized a cryptic variant that caused a breeding barrier in tomato. *Nature Plants* **5**, 471–479 (2019).

22. Chen, W. K., Swartz, J. D., Rush, L. J. & Alvarez, C. E. Mapping DNA structural variation in dogs. *Genome Res.* **19**, 500–509 (2009).

23. Guan, P. & Sung, W.-K. Structural variation detection using next-generation sequencing data. *Methods* **102**, 36–49 (2016).

24. Simpson, J. T. & Pop, M. The Theory and Practice of Genome Sequence Assembly. *Annu. Rev. Genomics Hum. Genet.* **16**, 153–172 (2015).

25. Jiao, W.-B. The impact of third generation genomic technologies on plant genome assembly. *Curr. Opin. Plant Biol.* **36**, 64–70 (2017).

26. Sedlazeck, F. J., Lee, H., Darby, C. A. & Schatz, M. C. Piercing the dark matter: Bioinformatics of long-range sequencing and mapping. *Nat. Rev. Genet.* **19**, 329–346 (2018).

27. Liu, S. *et al.* Discovery, genotyping and characterization of structural variation and novel sequence at single nucleotide resolution from de novo genome assemblies on a population scale. *Gigascience* **4**, (2015).

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

29. Kronenberg, Z. N. *et al.* High-resolution comparative analysis of great ape genomes. *Science* **360**, eaar6343 (2018).

30. Sedlazeck, F. J. *et al.* Accurate detection of complex structural variations using single-molecule sequencing. *Nat. Methods* **15**, 461–468 (2018).

31. Gong, L. *et al.* Picky comprehensively detects high-resolution structural variants in nanopore long reads. *Nat. Methods* **15**, 455–460 (2018).

32. Layer, R. M., Chiang, C., Quinlan, A. R. & Hall, I. M. LUMPY: a probabilistic framework for structural variant discovery. *Genome Biol.* **15**, R84 (2014).

33. Wijnker, E. *et al.* The genomic landscape of meiotic crossovers and gene conversions in Arabidopsis thaliana. *Elife* **2**, e01426 (2013).

34. Qi, J., Chen, Y., Copenhaver, G. P. & Ma, H. Detection of genomic variations and DNA polymorphisms and impact on analysis of meiotic recombination and genetic mapping. *Proc. Natl. Acad. Sci. U. S. A.* **111**, 10007–12 (2014).

35. Delcher, A. L. *et al.* Alignment of whole genomes. *Nucleic Acids Res.* **27**, 2369–2376 (1999).

36. Kurtz, S. *et al.* Versatile and open software for comparing large genomes. *Genome Biol.* **5**, R12 (2004).

37. Audano, P. A. *et al.* Characterizing the Major Structural Variant Alleles of the Human Genome. *Cell* **176**, 663-675.e19 (2019).

38. Chaisson, M. J. P. *et al.* Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nat. Commun.* **10**, 1784 (2019).

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

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

41. Coombe, L., Nikolić, V., Chu, J., Birol, I. & Warren, R. L. ntJoin: Fast and lightweight assembly-guided scaffolding using minimizer graphs. *Bioinformatics* btaa253 (2020).

42. Alonge, M. *et al.* RaGOO: fast and accurate reference-guided scaffolding of draft genomes. *Genome Biol.* **20**, 224 (2019).

43. International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. *Nature* **431**, 931–945 (2004).

44. Hirsch, C. N. *et al.* Draft Assembly of Elite Inbred Line PH207 Provides Insights into Genomic and Transcriptome Diversity in Maize. *Plant Cell* **28**, 2700–2714 (2016).

45. Jiao, Y. *et al.* Improved maize reference genome with single-molecule technologies. *Nature* **546**, 524 (2017).

46. Chakraborty, M. *et al.* Hidden genetic variation shapes the structure of functional elements in Drosophila. *Nat. Genet.* **50**, 20–25 (2018).

47. Hoskins, R. A. *et al.* The Release 6 reference sequence of the *Drosophila melanogaster* genome. *Genome Res.* **25**, 445–458 (2015).

48. Strope, P. K. *et al.* The 100-genomes strains, an S. cerevisiae resource that illuminates its natural phenotypic and genotypic variation and emergence as an opportunistic pathogen. *Genome Res.* **25**, 762–774 (2015).

49. Goffeau, A. *et al.* Life with 6000 Genes. *Science* **274**, 546–567 (1996).

50. Schnable, P. S. *et al.* The B73 maize genome: complexity, diversity, and dynamics. *Science* **326**, 1112–5 (2009).

51. Sun, H. *et al.* Linked-read sequencing of gametes allows efficient genome-wide analysis of meiotic recombination. *Nat. Commun.* **10**, 4310 (2019).

52. Rowan, B. A., Patel, V., Weigel, D. & Schneeberger, K. Rapid and Inexpensive Whole-Genome Genotyping-by-Sequencing for Crossover Localization and Fine-Scale Genetic Mapping. *G3 Genes, Genomes, Genet.* **5**, 385–398 (2015).

53. Jeffares, D. C. *et al.* Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. *Nat. Commun.* **8**, 14061 (2017).

54. Bartenhagen, C. & Dugas, M. RSVSim: an R/Bioconductor package for the simulation of structural variations. *Bioinformatics* **29**, 1679–1681 (2013).

55. Li, H. Wgsim: Reads simulator. *https://github.com/lh3/wgsim*

56. Michael, T. P. *et al.* High contiguity Arabidopsis thaliana genome assembly with a single nanopore flow cell. *Nat. Commun.* **9**, 541 (2018).

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

58. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–9 (2009).

59. Kiełbasa, S. M., Wan, R., Sato, K., Horton, P. & Frith, M. C. Adaptive seeds tame genomic sequence comparison. *Genome Res.* **21**, 487–493 (2011).

60. faTOFastq. Available at: http://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/.

61. Faust, G. G. & Hall, I. M. SAMBLASTER: fast duplicate marking and structural variant read extraction. *Bioinformatics* **30**, 2503–2505 (2014).

62. Smit, AFA, Hubley, R & Green, P. RepeatMasker Open-4.0. *http://www.repeatmasker.org*

63. Ossowski, S. *et al.* Sequencing of natural strains of Arabidopsis thaliana with short reads. *Genome Res.* **18**, 2024–33 (2008).

64. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B* **57**, 289–300 (1995).

65. The Computational Pan-genomics Consortium. Computational pan-genomics: status, promises and challenges. *Brief. Bioinform.* **19**, 118–135 (2018).

66. Paten, B., Novak, A. M., Eizenga, J. M. & Garrison, E. Genome graphs and the evolution of genome inference. *Genome Res.* **27**, 665–676 (2017).

67. Maretty, L. *et al.* Sequencing and de novo assembly of 150 genomes from Denmark

as a population reference. *Nat. Publ. Gr.* **548**, (2017).

68. Portwood, J. L. *et al.* MaizeGDB 2018: the maize multi-genome genetics and genomics database. *Nucleic Acids Res.* **47**, D1146–D1154 (2019).

6 Appendix

Parts of this chapter were the basis of a manuscript that was published as a peer-reviewed research article in Genome Biology (Goel et al. 2019)[1]*. However, in this chapter, I only discuss the work that was done by me. Results and data that were not generated by me are either clearly pointed or cited.*

Authors list (Goel et al. 2019): Manish Goel[α]*, Hequan Sun*[β]*, Wen-Biao Jiao*[α]*, Korbinian Schneeberger*[α, β]*.*

Author affiliations: [α]*Department of Chromosome Biology, Max Planck Institute for Plant Breeding Research, 50829 Cologne, Germany;* [β]*LMU Munich, 82152 Planegg-Martinsried, Germany.*

Authors contributions (Goel et al. 2019): The project was conceived by KS and WBJ. MG and KS developed the algorithms. MG implemented SyRI and performed all analyses. HS processed recombinant genome sequencing data and identified crossing-over sites. WBJ generated the Ler assembly. The manuscript was written by MG and KS with inputs from HS and WBJ. All authors read and approved the final manuscript.

6.1 Using SyRI and a working example

SyRI is an open-source method available under the MIT license and is available for download at github.com/schneebergerlab/syri. A user-manual is also available at schneebergerlab.github.io/syri/. I developed SyRI using Python3.5 and used Cython for computationally demanding tasks. SyRI was developed on Linux, but it can work on other platforms as well.

6.1.1 Installing SyRI

Pre-requisites

- C/C++ compiler: g++
- Python3.5
- Python packages: Cython, numpy, scipy, pandas, python-igraph, biopython, psutil, and pysam

Installing dependencies and SyRI:

- The python packages can be installed in a conda environment using:

```
conda install cython numpy scipy pandas biopython psutil
conda install -c conda-forge python-igraph
conda install -c bioconda pysam
```

- SyRI can be downloaded from github using:

```
git clone https://github.com/schneebergerlab/syri.git
```

- From the downloaded folder, SyRI can be installed using:

```
python3 setup.py install
chmod +x syri/bin/syri syri/bin/chroder       # Make scripts executable
```

6.1.2 Working example

Here, I provide a small pipeline that can be used to download, pre-process and analyse two yeast genomes using SyRI. For the purpose of this example, I assume that minimap2 is installed and in the working path of the user[57].

- Step 1: From within the working directory, download the yeast genomes from NCBI:

```
## Get Yeast Reference genome
wget
ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/146/045/GCA_000146045.2_R
64/GCA_000146045.2_R64_genomic.fna.gz

## Get Query genome
wget
ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/977/955/GCA_000977955.2_S
c_YJM1447_v1/GCA_000977955.2_Sc_YJM1447_v1_genomic.fna.gz
```

- Step 2: Unzip the genomes and remove mitochondrial chromosomes:

```
gzip -df GCA_000146045.2_R64_genomic.fna.gz
gzip -df GCA_000977955.2_Sc_YJM1447_v1_genomic.fna.gz
```

```
## Remove mitochondrial DNA
head -151797 GCA_000977955.2_Sc_YJM1447_v1_genomic.fna >
GCA_000977955.2_Sc_YJM1447_v1_genomic.fna.filtered

## Create symlink objects
ln -sf GCA_000146045.2_R64_genomic.fna refgenome
ln -sf GCA_000977955.2_Sc_YJM1447_v1_genomic.fna.filtered qrygenome
```

- Step 3: Perform whole-genome alignment using minimap2:

```
## Using minimap2 for generating alignment
minimap2 -ax asm5 --eqx refgenome qrygenome > out.sam
samtools view -b out.sam > out.bam
```

- Step 4: Identify genomic differences using SyRI:

```
python3 $PATH_TO_SYRI -k -F B -c out.bam -r refgenome -q
qrygenome
```

6.2 Additional Tables:

Table S1: Statistics of assemblies used.

Species	Sample/Strain	NCBI Accession	Assembly size (in Mb)	Sequence information (% genome size)
			After pre-processing	
Arabidopsis thaliana	Reference/Col-0	GCA_000001735.3	119.1	0.99
Arabidopsis thaliana	Ler	GCA_900660825	118.0	0.99
Homo sapiens	Reference	GCA_000001405.27	3088.2	0.95
Homo sapiens	NA12878	GCA_002077035.3	3034.9	0.92
Homo sapiens	NA19240	GCA_001524155.4	3037.3	0.92
Saccharomyces cerevisiae	Reference/S288C	GCA_000146045.2	12.0	1.00
Saccharomyces cerevisiae	YJM1447	GCA_000977955.2	12.1	0.99
Drosophila melanogaster	Reference/iso-1	GCA_000001215.4	133.8	0.99
Drosophila melanogaster	A4	GCA_002300595.1	135.5	0.99
Zea Mays	Reference/B73	GCA_000005005.6	2106.3	0.19
Zea Mays	PH207	GCA_002237485.1	2060.2	0.18

(from Goel *et al.*, 2019)

Table S2: Number of large (>1Mb) variations identified by each method in L*er* genome.

Method	Number of deletions	Number of tandem duplications
SyRI	0	0
AsmVar	0	0
Assemblytics	0	0
Sniffles_PB	38	26
Sniffles_ONT	49	39
Pickt_PB	30	47
Picky_ONT	32	37
LUMPY	155	149

(from Goel *et al.*, 2019)

Table S3: Whole-genome alignment commands used.

Species	Genome A	Genome B	Command
Arabidopsis thaliana	Col-0	Ler	nucmer --maxmatch -c 100 -b 500 -l 50 refgenome qrygenome delta-filter -m -i 90 -l 100 out.delta > out_m_i90_l100.delta show-coords -THrd out_m_i90_l100.delta > out_m_i90_l100.coords
Homo sapiens	GRCh39.p12	NA12878	nucmer --maxmatch -c 500 -b 500 -l 100 refgenome qrygenome delta-filter -m -i 90 -l 100 out.delta > out_m_i90_l100.delta show-coords -THrd out_m_i90_l100.delta > out_m_i90_l100.coords
Homo sapiens	GRCh39.p12	NA19240	nucmer --maxmatch -c 500 -b 500 -l 100 refgenome qrygenome delta-filter -m -i 90 -l 100 out.delta > out_m_i90_l100.delta show-coords -THrd out_m_i90_l100.delta > out_m_i90_l100.coords
Saccharomyces cerevisiae	S288C	YJM1447	nucmer --maxmatch -c 100 -b 500 -l 50 refgenome qrygenome delta-filter -m -i 90 -l 100 out.delta > out_m_i90_l100.delta show-coords -THrd out_m_i90_l100.delta > out_m_i90_l100.coords
Drosophila melanogaster	iso-1	A4	nucmer --maxmatch -c 100 -b 500 -l 50 refgenome qrygenome delta-filter -m -i 90 -l 100 out.delta > out_m_i90_l100.delta show-coords -THrd out_m_i90_l100.delta > out_m_i90_l100.coords
Zea Mays	B73	PH207	nucmer --maxmatch -c 500 -b 500 -l 100 refgenome qrygenome delta-filter -m -i 90 -l 100 out.delta > out_m_i90_l100.delta show-coords -THrd out_m_i90_l100.delta > out_m_i90_l100.coords

(from Goel *et al.,* 2019)

6.3 Additional Figures

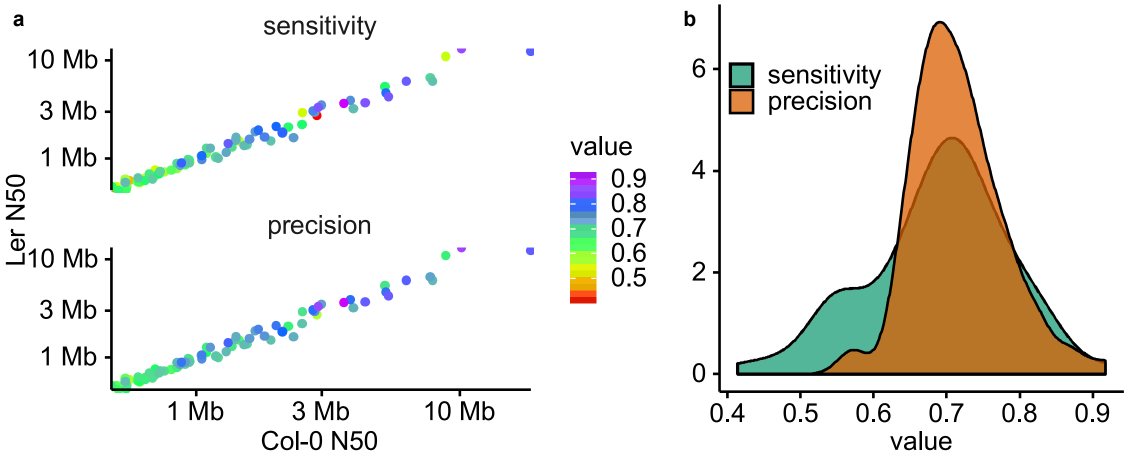


Figure S1: SyRI's rearrangement performance when both assemblies are incomplete. (a) Points represent samples, colour represents sensitivity and precision values. The x- and y-axes represent N50 values for Col-0 and L*er* incomplete assemblies respectively. (b) Distribution of sensitivity and precision values across samples. (from Goel et al., 2019)

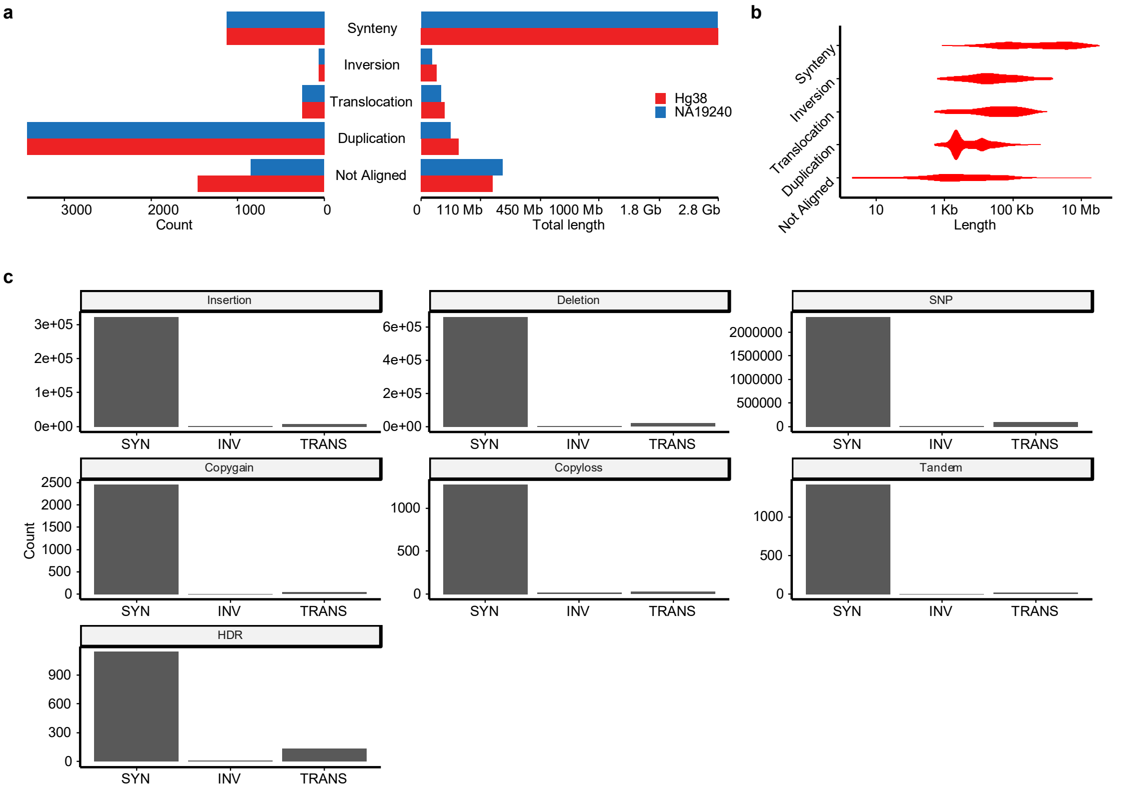


Figure S2: Genomic differences between human reference genome (Hg38) and genome NA19240. (a) Total length and number of syntenic, structurally rearranged, and not-aligned regions identified in the two genomes. (b) Length distribution of predicted regions. (c) The number of sequence variations found. HDR: highly diverged regions. (from Goel et al., 2019)

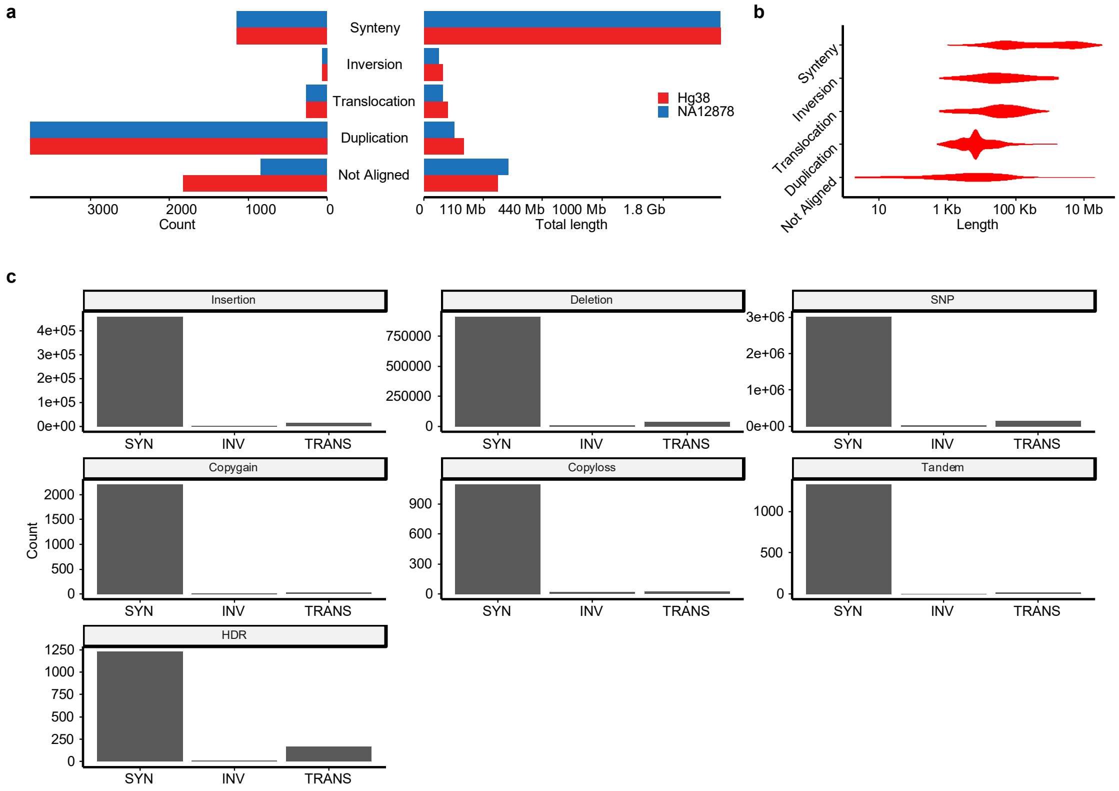


Figure S3: Genomic differences between human reference genome (Hg38) and genome NA12878. (a) Total length and number of syntenic, structurally rearranged, and not-aligned regions identified in the two genomes. (b) Length distribution of predicted regions. (c) The number of sequence variations found. HDR: highly diverged regions. (from Goel et al., 2019)

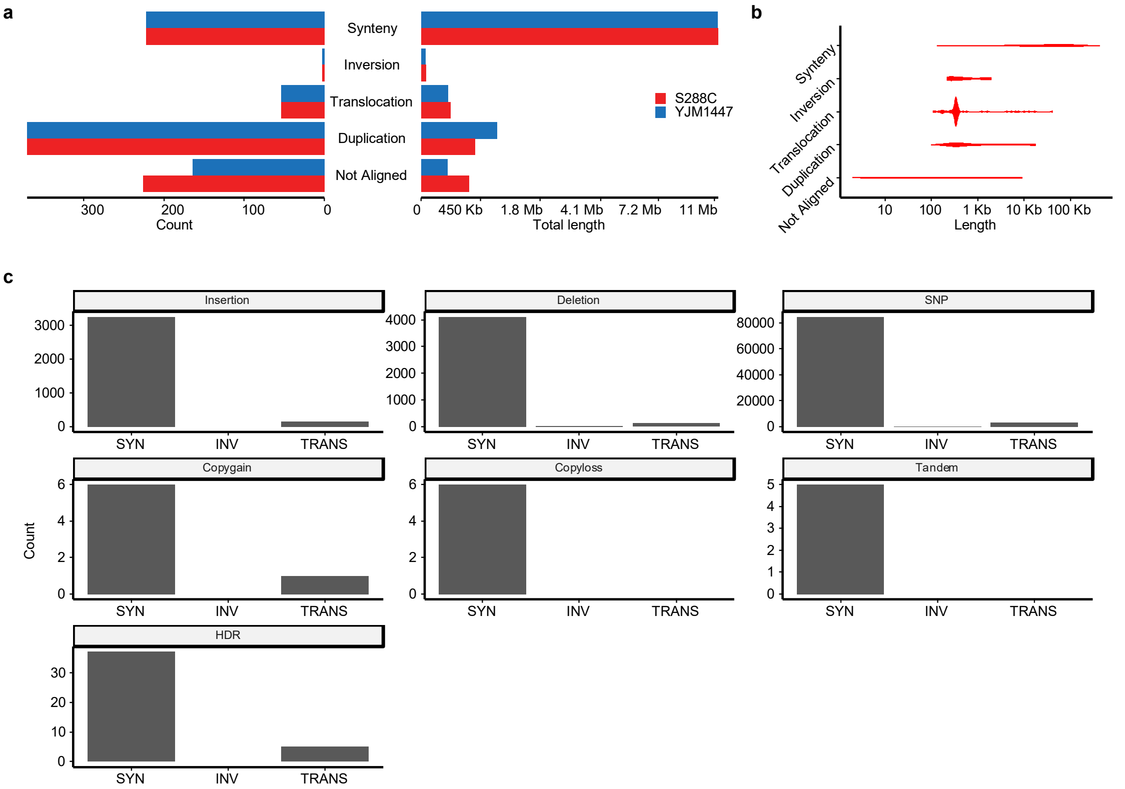


Figure S4: Genomic differences between yeast reference genome (S288C) and accession YJM1447. (a) Total length and number of syntenic, structurally rearranged, and not-aligned regions identified in the two genomes. (b) Length distribution of predicted regions. (c) The number of sequence variations found. HDR: highly diverged regions. (from Goel et al., 2019)

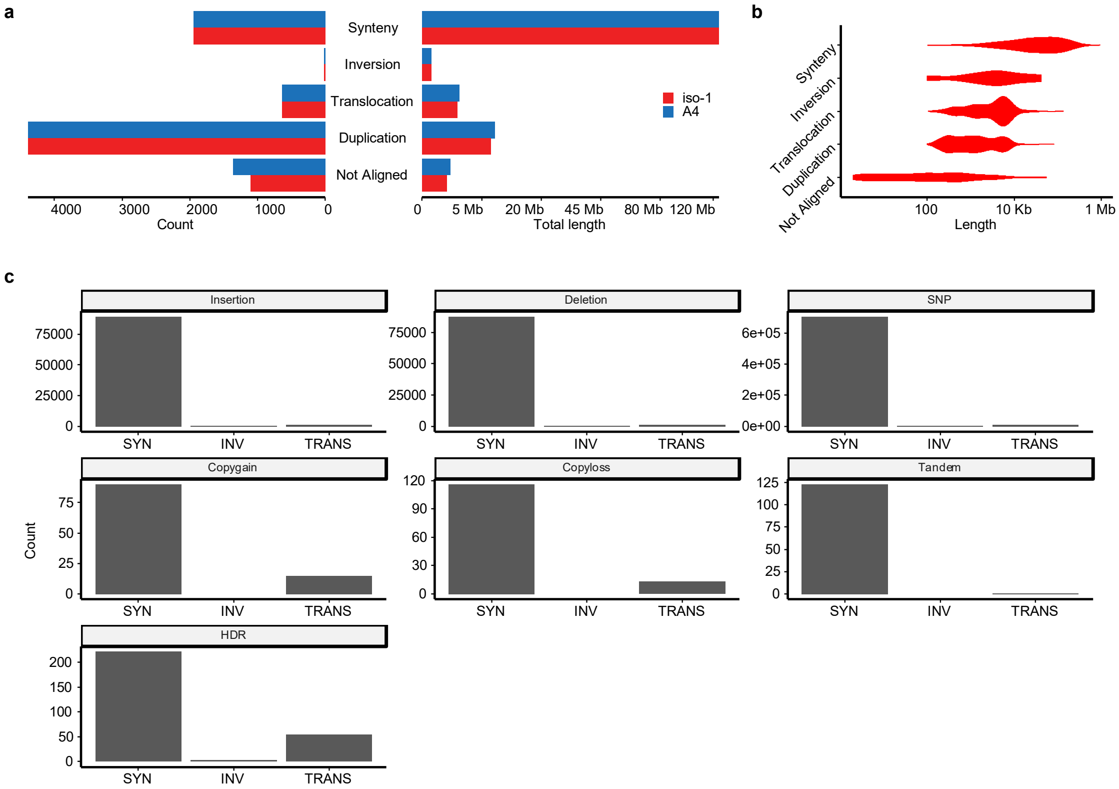


Figure S5: Genomic differences between fruit-fly reference genome (iso-1) and accession A4. (a) Total length and number of syntenic, structurally rearranged, and not-aligned regions identified in the two genomes. (b) Length distribution of predicted regions. (c) The number of sequence variations found. HDR: highly diverged regions. (from Goel *et al.*, 2019)

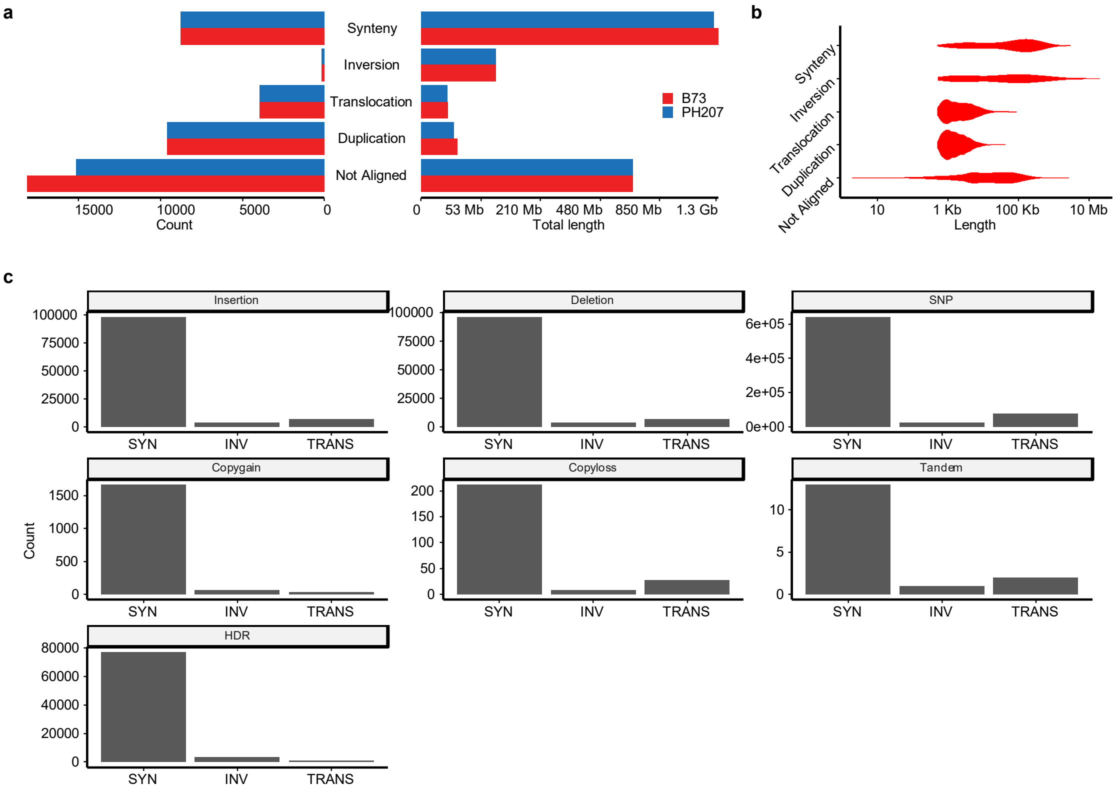


Figure S6: Genomic differences between maize reference genome (B73) and accession PH207. (a) Total length and number of syntenic, structurally rearranged, and not-aligned regions identified in the two genomes. (b) Length distribution of predicted regions. (c) The number of sequence variations found. HDR: highly diverged regions. (from Goel *et al.*, 2019)

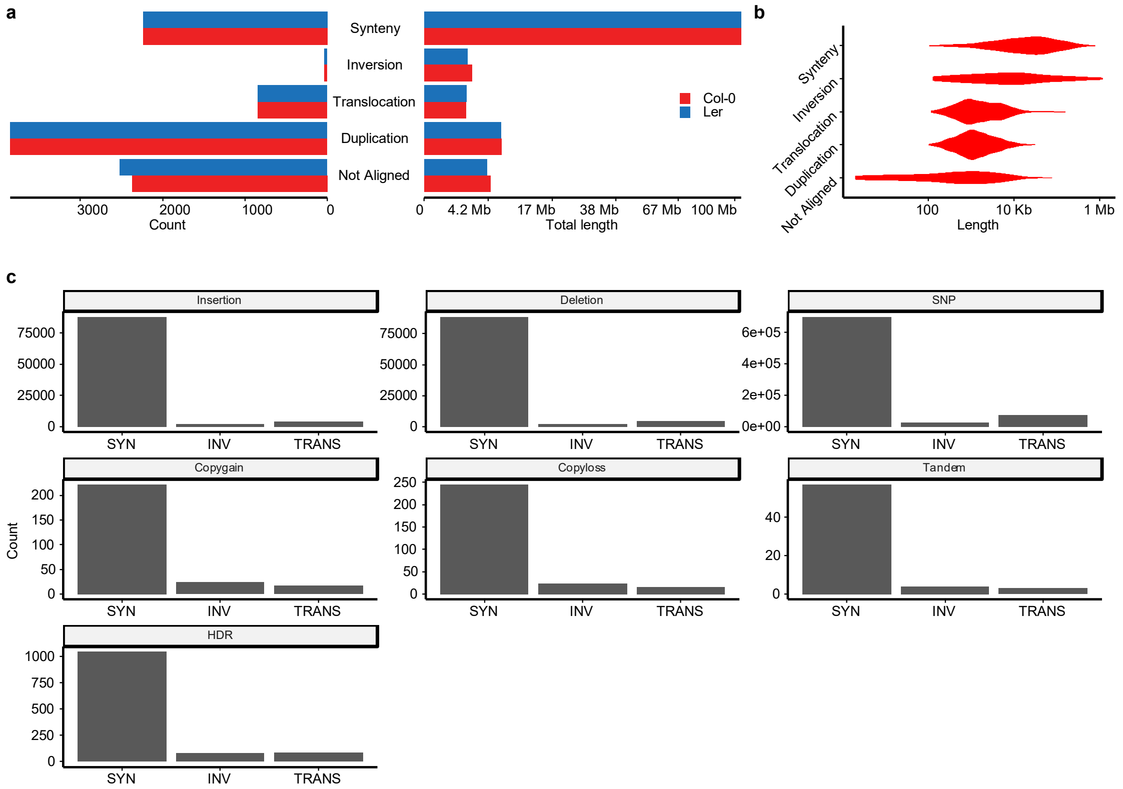


Figure S7: Genomic differences between *A. thaliana* reference genome (Col-0) and accession L*er*. (a) Total length and number of syntenic, structurally rearranged, and not-aligned regions identified in the two genomes. (b) Length distribution of predicted regions. (c) The number of sequence variations found. HDR: highly diverged regions. (from Goel *et al.*, 2019)

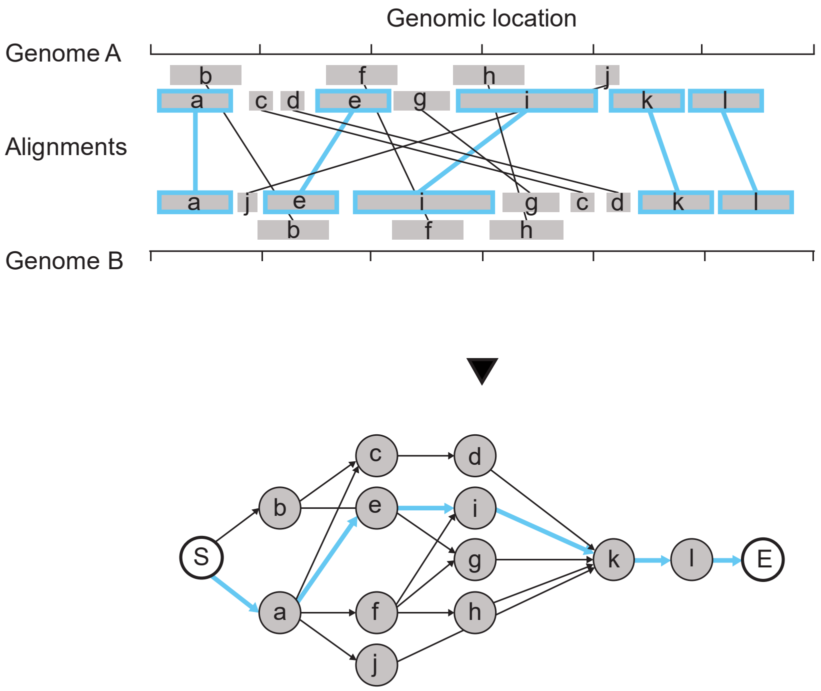


Figure S8: Using a Directed Acyclic Graph (DAG) for syntenic path identification. Grey blocks represent alignments between Genome A and Genome B (top). Overlapping alignments are stacked vertically ("b" is overlapping "a" on Genome A and it is overlapping "e" on Genome B). Using alignments, a directed acyclic graph (DAG) is generated (bottom). The grey nodes correspond to the alignments and the white nodes ("S" and "E") represent imaginary nodes. Longest path in the graph is shown by the blue line, corresponding alignments of the syntenic path have the blue boundary. (from Goel *et al.*, 2019)

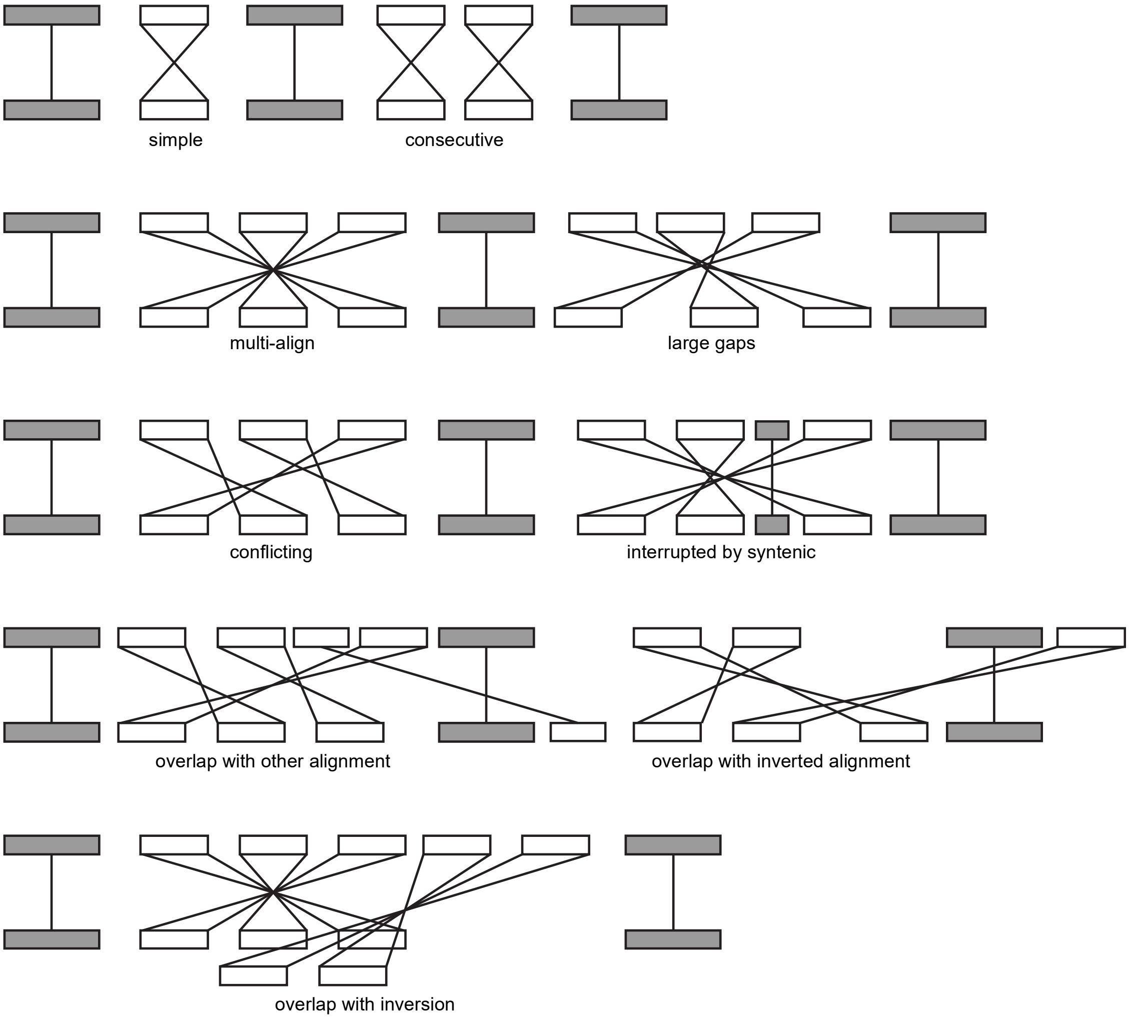

simple
consecutive
multi-align
large gaps
conflicting
interrupted by syntenic
overlap with other alignment
overlap with inverted alignment
overlap with inversion

Figure S9: Various conformation of inversions and inverted alignments. White and grey blocks correspond to inverted and syntenic alignments, respectively. Inversions consist of inverted alignments surrounded by syntenic alignments on both sides. An inversion can consist of a simple inverted alignment or multiple inverted alignments. More than one inversions can occur consecutively. Inverted alignments in a region could represent conflicting inversions and can be overlapping with other alignments. Large inversions could be interrupted by alignments from TDs or syntenic regions. (from Goel *et al.*, 2019)

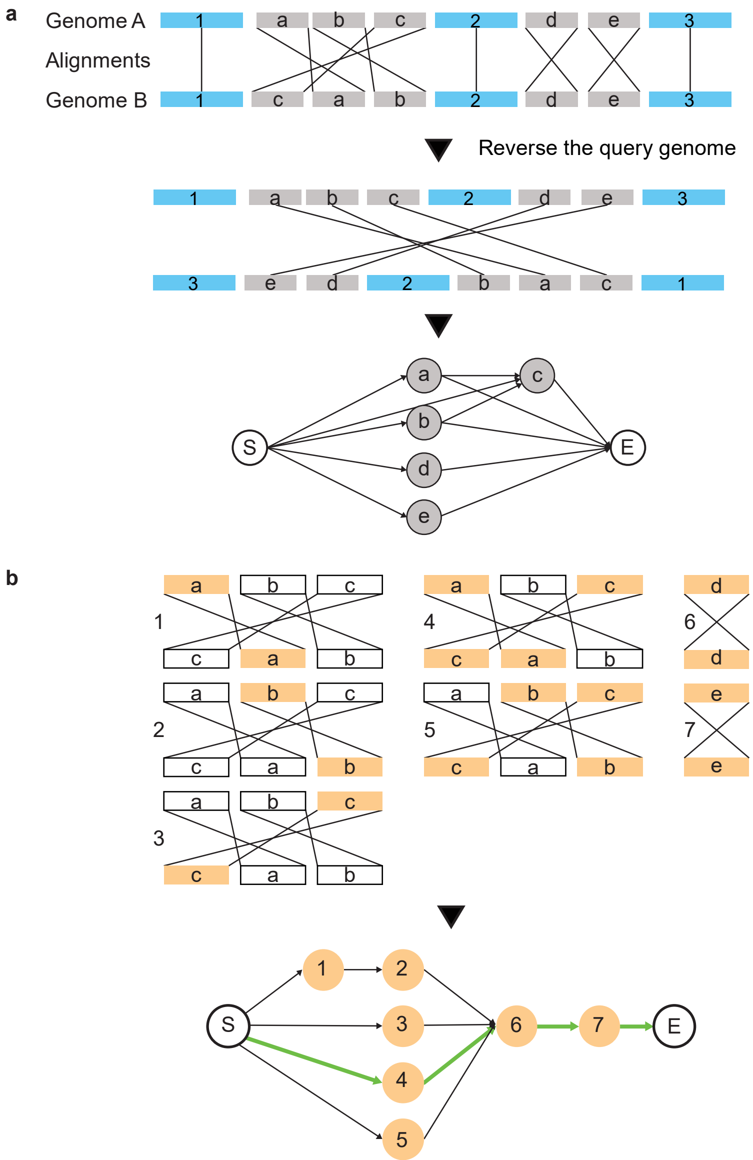
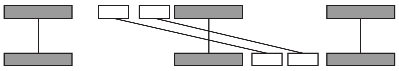
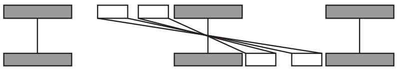


Figure S10: Two-step method for inversion identification. (a) Grey and blue blocks correspond to inverted and already annotated syntenic alignments, respectively (top). A directed acyclic graph (DAG) is generated from the originally inverted alignments (bottom). Grey and white (S, E) nodes correspond to alignments and imaginary nodes, respectively. Each S→E path represents a candidate inversion. (b) In this case, seven candidates are identified. Orange blocks represent inverted alignments that constitute the candidate inversion. A second DAG is generated using where orange nodes correspond to the candidates, while white nodes ("S" and "E") are imaginary nodes. The green line is the highest scoring path from node S to E, and it constitutes of the highest scoring non-conflicting inversions. (from Goel *et al.*, 2019)

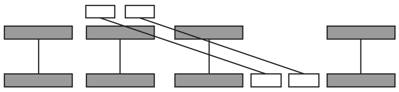
a)



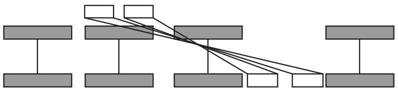
One translocation consisting of two alignments



One inverted translocation consisting of two alignments

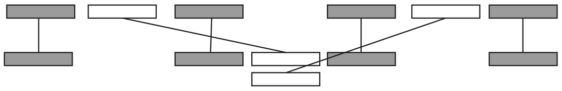


One duplication consisting of two alignments

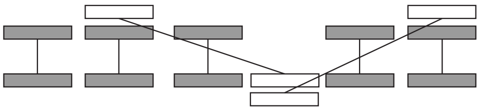


One inverted duplication consisting of two alignments

b)

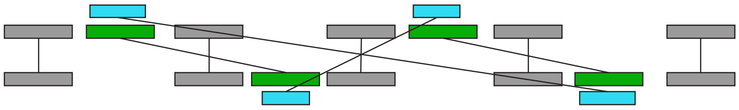


Conflicting annotations: Only one candidate TD can be selected as translocation. Other need to be annotated as duplication.

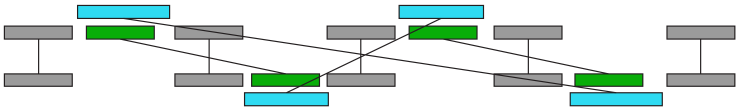


Conflicting annotations: Only one candidate can be selected as duplication. Other need to be defined as redundant and removed.

c)



Network of rearranged regions: Green candidates would be selected as translocations and cyan candidates would be removed as redundant.



Network of rearranged regions: Cyan candidates would be selected as translocations and green candidates would be removed as redundant.

Figure S11: Complexities of translocations and duplication identification. Grey blocks represent syntenic/inverted regions. (a) Examples of TDs consisting of more than one alignment. White blocks are alignments (directed or inverted). (b) White blocks represent candidate TDs (a group of alignments representing one TD). The candidate TDs can overlap each other resulting in conflicting annotations. Here, the rearranged region in the lower genome can be annotated as translocation (or duplication) from two different regions resulting in conflicting annotations. (c) Green and cyan blocks represent candidate TDs that form a network of overlapping candidates. (from Goel *et al.*, 2019)

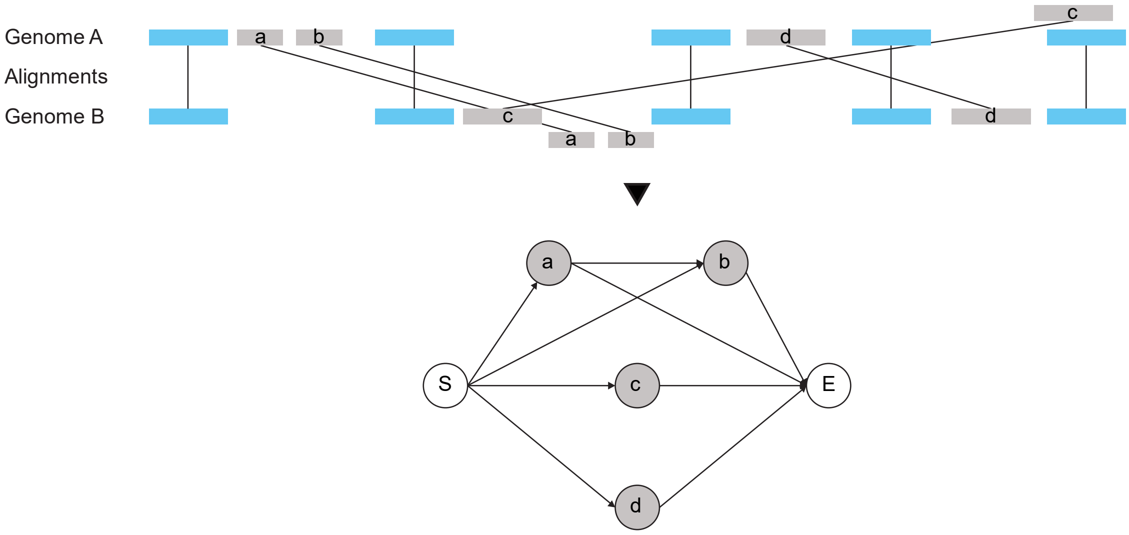


Figure S12: Using a Directed Acyclic Graph (DAG) for candidate TD identification. Grey and blue blocks are currently un-annotated alignments and already annotated syntenic/inverted regions, respectively. In the DAG, each node represents an alignment and edges are added between nodes that can together represent a TD. Two imaginary nodes (S and E) are added (white nodes) and edges are added from node S to all other nodes and from all other nodes to node E. Each S→E path corresponds to a candidate TD. (from Goel *et al.*, 2019)

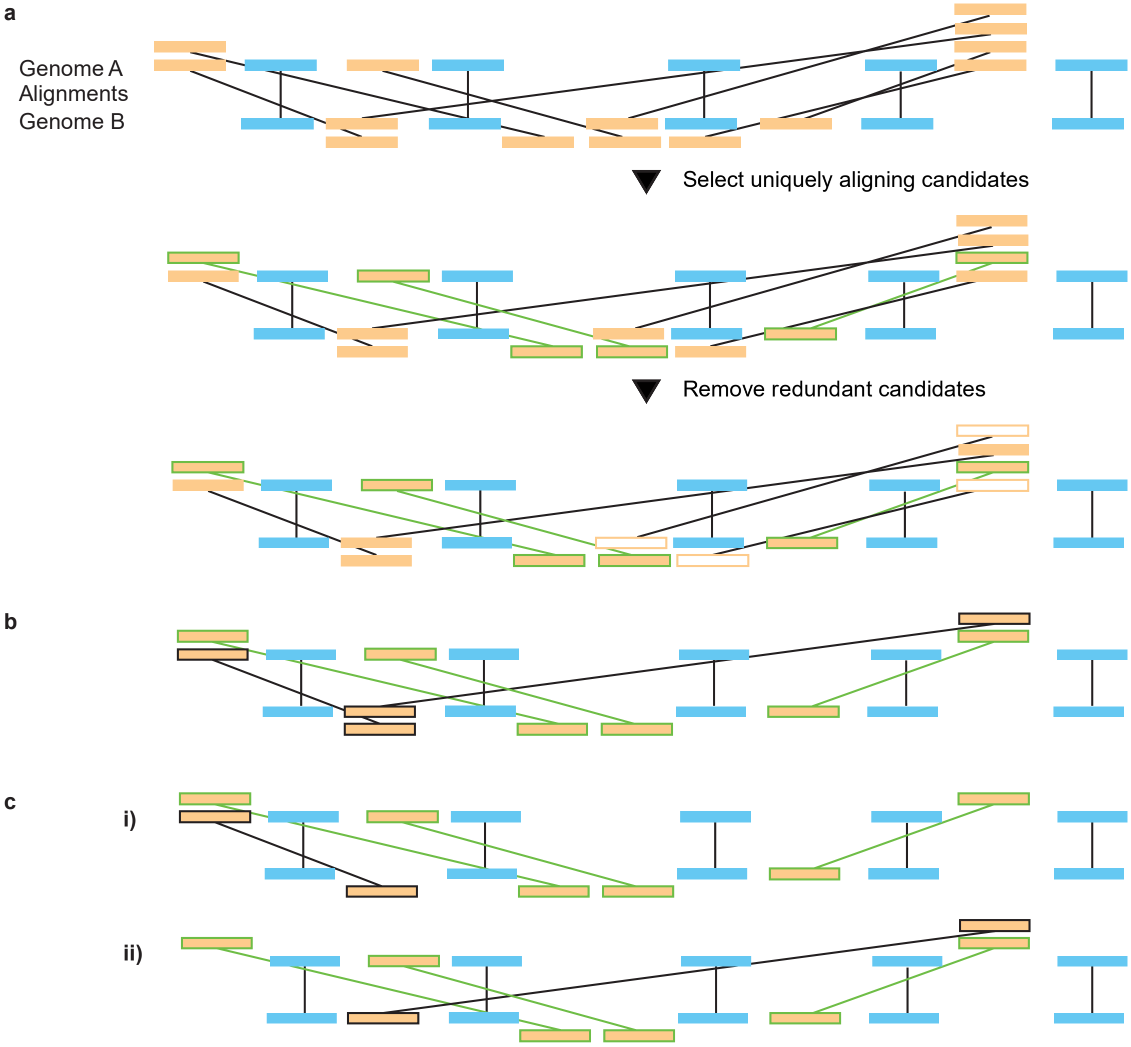

a
Genome A
Alignments
Genome B
Select uniquely aligning candidates
Remove redundant candidates
b
c
i)
ii)

Figure S13: Selecting non-conflicting candidate TDs. a) A network of overlapping candidate TDs (orange blocks) and syntenic/inverted regions (blue alignments). Seven candidates TDs align three regions on genome A to five regions on genome B. Candidates that align a region uniquely (green boundary candidates) are necessary candidates and will be part of the output group of candidates. Candidates that align already annotated regions (white candidates) are redundant and removed. b) Deadlock is when no more necessary or redundant candidates can be selected (black boundary). c) Deadlocks in smaller networks are solved using brute-force by listing all possible combinations in which the remaining candidates can be selected (i and ii). The highest-scoring combination is selected as the output. (from Goel *et al.*, 2019)

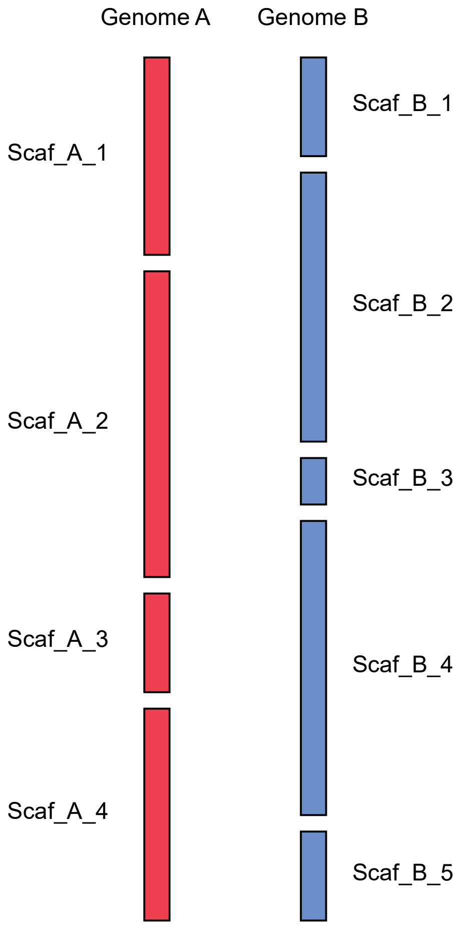

Genome A
Genome B
Scaf_A_1
Scaf_A_2
Scaf_A_3
Scaf_A_4
Scaf_B_1
Scaf_B_2
Scaf_B_3
Scaf_B_4
Scaf_B_5

Figure S14: Homology based pseudo-chromosome generation. Red and blue lines show scaffolds/contigs from Genome A and Genome B, respectively. Smaller scaffolds/contigs from one genome aligning to the same scaffolds/contigs in the other genome are grouped and ordered.

7 Erklärung zur Dissertation

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie – abgesehen von unten angegebenen Teilpublikationen – noch nicht veröffentlicht worden ist, sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde.

Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. Korbinian Schneeberger betreut worden.

Ich versichere, dass ich alle Angaben wahrheitsgemäß nach bestem Wissen und Gewissen gemacht habe und verpflichte mich, jedmögliche, die obigen Angaben betreffenden Veränderungen, dem Dekanat unverzüglich mitzuteilen

Teilpublikationen:

1) **Goel, M.**, Sun, H., Jiao, W. *et al.* SyRI: finding genomic rearrangements and local sequence differences from whole-genome assemblies. *Genome Biol* **20,** 277 (2019). https://doi.org/10.1186/s13059-019-1911-0

Manish Goel

Digitally signed by Manish Goel
DN: cn=Manish Goel, o=MPIPZ, ou, email=mnshgl0110@gmail.com, c=DE
Date: 2020.06.30 13:46:04 +02'00'

Datum / Unterschrift