Single-cell strand sequencing for structural variant analysis and genome assembly

Dissertation submitted towards the degree
Doctor of Natural Sciences
of the Faculty of Mathematics and Computer Science
of Saarland University

by

Maryam Ghareghani

Saarbrücken
2022

Colloquium date: 02.11.2022

Dean: Univ.-Prof. Dr. Jürgen Steimle

Chairman: Prof. Dr. Volkhard Helms

Reviewers:
Prof. Dr. Tobias Marschall
Prof. Dr. Sven Rahmann
Prof. Dr. Martin Vingron

Scientific assistant: Dr. Alexander Gress

Abstract

Rapid advances of DNA sequencing technologies and development of computational tools to analyze sequencing data has started a revolution in the field of genetics. DNA sequencing has applications in medical research, disease diagnosis and treatment, and population genetic studies. Different sequencing techniques have their own advantages and limitations, and they can be used together to solve genome assembly and genetic variant detection.

The focus of this thesis is on a specific single-cell sequencing technology, called strand sequencing. With its chromosome and haplotype-specific strand information, this technique has very powerful signals for discovery of genomic structural variations, haplotype phasing, and chromosome clustering. We developed statistical and computational tools to exploit this information from strand sequencing technology.

I first present a computational framework for detecting structural variations in single cells using strand sequencing data. The presented tool is able to detect different types of structural variations in single cells including copy number variations, inversions, and inverted duplications, and also more complex biological events such as translocations and breakage-fusion-bridge (BFB) cycles. These variations and genomic rearrangements have been observed in cancer, therefore the discovery of such events within cell populations can lead to a more accurate picture of cancer genomes and help in diagnosis.

In the remainder of this thesis, I elaborate on two computational pipelines for clustering long DNA sequences by their original chromosome and haplotype in the absence of a reference genome. These pipelines are developed to facilitate genome assembly and *de novo* haplotype phasing in a fast and accurate manner. The resulting haplotype assemblies can be useful in studying genomic variations with no reference bias, gaining insights in population genetics, and detection of compound heterozygosity.

Kurzfassung

Die rasanten Fortschritte im Bereich der DNA-Sequenzierung und die Entwicklung von Computerwerkzeugen für die Analyse von Sequenzierdaten haben eine Revolution auf dem Gebiet der Genetik ausgelöst. Die DNA-Sequenzierung findet Anwendung in der medizinischen Forschung, bei der Diagnose und Behandlung von Krankheiten und bei populationsgenetischen Studien. Verschiedene Sequenzierungstechniken haben jeweils ihre Vorteile und Grenzen, können aber kombiniert werden, um Genome zu assemblieren oder um genetische Varianten zu finden.

Der Schwerpunkt dieser Arbeit liegt auf einer speziellen Einzelzell Sequenzierungstechnologie, genannt Strand-Seq. Mit ihren chromosomen- und haplotypspezifischen Stranginformationen liefert diese Technik sehr starke Signale für die Entdeckung genomischer Strukturvariationen, die Rekonstruktion von Haplotypen und das Chromosomenclustering. Wir haben statistische und computergestützte Werkzeuge entwickelt, um diese Informationen der Strand-Seq Technologie zu nutzen.

Zunächst präsentiere ich einen mathematisches Modell für die Erkennung struktureller Variationen in einzelnen Zellen unter Verwendung von Strand-Seq Daten. Das vorgestellte Tool ist in der Lage, verschiedene Arten von Strukturvariationen in Einzelzellen zu erkennen, darunter Kopienzahlvariationen, Inversionen und invertierte Duplikationen sowie komplexere biologische Ereignisse wie Translokationen und Break-Fusion-Bridge-Zyklen (BFB). Diese Variationen und genomischen Umlagerungen wurden bei Krebs beobachtet, sodass der Nachweis solcher Ereignisse in Zellpopulationen zu einem genaueren Bild des Krebsgenoms führen und bei der Diagnose helfen kann.

Im Folgenden stelle ich zwei Computerpipelines vor, mit denen lange DNA-Sequenzen nach ihrem ursprünglichen Chromosom und Haplotyp geclustert werden können, wenn kein Referenzgenom verfügbar ist. Diese Pipelines wurden entwickelt, um die Genomassemblierung und die *de novo* Rekonstruktion von Haplotypen auf schnelle und genaue Weise zu erleichtern. Die daraus resultierenden Haplotypen können bei der Untersuchung genomischer Variationen ohne Referenzverzerrung, bei der Gewinnung von Einblicken in die Populationsgenetik und beim Nachweis von zusammengesetzter Heterozygotie nützlich sein.

Acknowledgments

I would like to thank my supervisor, Tobias Marschall, for his great help and support during my PhD. I also thank all my family and friends who accompanied me during my PhD journey. Special thanks to my good friends and colleagues Mikko Rautiainen, Rebecca Serra Mari, Jana Ebler, David Porubsky, Ali Ghaffari, Aryan Kamal, Hufsah Ashraf, Fawaz Dabbaghieh, and Peter Ebert for creating a friendly and lovely atmosphere at work. Last but not the least, great thanks to my family, specially my parents, who have been always supporting me during all stages of my life.

I dedicate this thesis to my parents who always supported me in my life.

Contents

Chapter 1

Introduction

DNA is the molecule of inheritance responsible for creating different forms of life. It exists in all living organisms and carries essential genetic information for their reproduction, function, and development. A genome[1] includes genes that encode proteins, which are large bio-molecules performing a wide range of functions in living organisms. Proteins play a vital role in metabolism, cell signaling, immune responses, and many other essential tasks in the body.

Mutations in the genome can alter genes and their resulting proteins and create various phenotypes or diseases in the body. According to Darwin's theory of natural selection [18], these variations can fit to the environment and survive in a population or be less successful and go extinct. Some mutations could lead to significant damages to the DNA and cause cancer. The study of the genome is hence very important to discover the genetic traits in a population and finding mutations which could help in discovering disease mechanisms and treatment.

The connection between DNA, genes and proteins was not initially discovered in classical genetics. The term *genes* was first scientifically introduced by Mendel as units of inheritance that can transmit genetic traits from parents to offspring. In his pea plants experiments, Mendel formalized the common patterns of inheritance as laws of random segregation and assortment [71]. He discovered that different alleles of a gene[2] are inherited independently from parents to their offspring.

Classical genetics was able to explain inheritance patterns in visible traits. However, the molecular mechanisms of genes and their expression, mutation, and reproduction was still unknown until the emergence of molecular biology, a new field of science that established the connection between DNA, genes, and proteins.

The central dogma of molecular biology, first stated by Francis Crick, declares the flow of information from DNA to RNA and then to proteins. The term *information* was first used by James Watson and Francis Crick who published the double helical structure of DNA [111]. DNA had been discovered to consist of two strands of complementary nucleotide base pairs that bind together by hydrogen bonds. Watson and Crick mentioned DNA as an informational molecule: "It therefore seems likely that the precise sequence of the bases is the code which carries the genetic information" [111].

The discoveries in molecular biology motivated scientists to determine the exact

[1] the full content of DNA

[2] type of a gene

order of the bases in the genome, known as DNA sequencing problem. The first sequencing technology was invented for protein sequencing by Fredrick Sanger. He later developed a DNA sequencing technology with the chain termination method, based on the state of the art primer-extension strategy [97]. He was one of the few scientists who won two Nobel prizes, one for sequencing protein and the other one for sequencing DNA. Due to its ease and efficiency at the starting time of the human genome project in 1990, Sanger sequencing technology became the method of choice in this project that led to the publication of the first human genome sequence at 2001.

At the beginning of the 21st century, next generation sequencing (NGS) technologies were developed and implemented commercially. These technologies used massively parallel sequencing methods that reduced the sequencing cost and time to orders of magnitude lower compared to the traditional Sanger sequencing method. The advances in sequencing technologies continued to the development of third generation and single-cell sequencing technologies. Nowadays, there is a broad range of existing sequencing technologies with their specific powers and limitations that are complementary to each other and can be used together to tackle several biological questions.

Thanks to the technological advances, there is a huge amount of sequencing data that needs to be analyzed with computational, mathematical and statistical methods to solve the related biological questions. Computational methods in genomics can be utilized in error correction of sequencing data, genome assembly, comparative genomics, linking genome mutations to diseases, studies of evolutionary biology, population genetics studies, and many other applications.

The focus of this thesis is on a single-cell sequencing technology named strand sequencing (Strand-seq). This technology has strong signals for large scale genomic variations, namely structural variants, and haplotype phasing. In the second chapter of this thesis, I present a computational pipeline for the discovery of structural variants in single cells. Structural variants are among the main classes of driver mutations for cancer, and detecting them at the level of single cells can give us an informative picture over the genome structure and its heterogeneity in cell population. The third chapter of the thesis presents an EM clustering algorithm for clustering long third generation sequencing reads by their original chromosome and sequencing direction. The computational workflow for generalization of the clustering method to further cluster long DNA sequences by their original haplotype is proposed in the forth chapter. The aforementioned clustering tasks can improve genome assembly in terms of accuracy, required computational resources, runtime and parallelization.

Publications

The main projects that I was involved in during my PhD studies include development of the scTRIP pipeline for single-cell structural variant detection, the SaaRclust tool for chromosome clustering of long sequencing reads, and the Haploclust tool for haplotype clustering of unitigs in overlap graphs. The article of SaaRclust has been accepted at the ISMB conference in 2018, which is the biggest annual conference in Bioinformatics. It was published at the proceeding section of ISMB in Bioinformatics [37]. The scTRIP computational pipeline for single-cell structural variant detection was published in

Nature Biotechnology in 2019 [95]. The Haploclust pipeline is still an ongoing project and will be submitted to a related journal after finalizing the results.

The projects in which I contributed as co-author include a fully phased assembly of a human genome [88] and the Human Genome Structural Variation Consortium 2 (HGSVC2) [23]. The phased assembly pipeline was published in Nature Biotechnology [88]. It was one of the first assembly pipelines providing a contiguous chromosome-scale haplotype assembly of a human diploid genome without using the parental data. The Human Genome Structural Variation Consortium 2 (HGSVC2), published in Science [23], provided 64 human genome assemblies based on the aforementioned assembly pipeline. HGSVC is an international consortium following up the work of the 1000 Genomes Project structural variation analysis group. It aims to define a high quality map of genome structural variations in the human population. The second phase of this consortium (HGSVC2) led to assembly of haplotype-resolved human genomes from different populations together with the detection and functional analysis of structural variants. This study generated an accurate panel of human reference genomes providing insights into the diversity of human populations in terms of segregating structural variants.

The SaaRclust method was used in the core genome assembly pipeline in the aforementioned projects [23, 88]. The inversion detection method in our scTRIP pipeline was also adjusted and used in HGSVC2 for analyzing inversions. I was a part of the inversion detection subgroup at HGSVC2, where my main contribution was to adjust the scTRIP pipeline to get the set of SV breakpoints resulting from multiple different sequencing technologies as input for the inversion detection task. The ongoing Haploclust project aims to improve the haplotype assembly method utilized in [88] with the idea of phasing DNA sequences directly on the assembly graph.

My PhD publications, including the aforementioned main papers and the research articles in which I contributed as co-author, are listed below (first authors' names are marked with *):

- *Strand-seq enables reliable separation of long reads by chromosome via expectation maximization.* **Maryam Ghareghani***, David Porubsky*, Ashley D Sanders, Sascha Meiers, Evan E Eichler, Jan O Korbel, Tobias Marschall (2018). Bioinformatics (Proceedings of ISMB), 34(13):i115-23.

- *Single-cell analysis of structural variations and complex rearrangements with tri-channel processing.* Ashley D. Sanders*, Sascha Meiers*, **Maryam Ghareghani***, David Porubsky*, Hyobin Jeong, M. Alexandra C. C. van Vliet, Tobias Rausch, Paulina Richter-Pechańska, Joachim B. Kunz, Silvia Jenni, Davide Bolognini, Gabriel M. C. Longo, Benjamin Raeder, Venla Kinanen, Jürgen Zimmermann, Vladimir Benes, Martin Schrappe, Balca R. Mardin, Andreas E. Kulozik, Beat Bornhauser, Jean-Pierre Bourquin, Tobias Marschall & Jan O. Korbel (2020). Nature Biotechnology, 38(3), 343-354.

- *Fully phased human genome assembly without parental data using single-cell strand sequencing and long reads.* David Porubsky*, Peter Ebert*, Peter A. Audano, Mitchell R. Vollger, William T. Harvey, Pierre Marijon, Jana Ebler, Katherine M. Munson, Melanie Sorensen, Arvis Sulovari, Marina Haukness, **Maryam Ghareghani**, Human

Genome Structural Variation Consortium, Peter M. Lansdorp, Benedict Paten, Scott E. Devine, Ashley D. Sanders, Charles Lee, Mark J. P. Chaisson, Jan O. Korbel, Evan E. Eichler & Tobias Marschall (2021). Nature Biotechnology, 39(3), 302-308.

- *Haplotype-resolved diverse human genomes and integrated analysis of structural variation.* Peter Ebert*, Peter A. Audano*, Qihui Zhu*, Bernardo Rodriguez-Martin*, David Porubsky, Marc Jan Bonder, Arvis Sulovari, Jana Ebler, Weichen Zhou, Rebecca Serra Mari, Feyza Yilmaz, Xuefang Zhao, PingHsun Hsieh, Joyce Lee, Sushant Kumar, Jiadong Lin, Tobias Rausch, Yu Chen, Jingwen Ren, Martin Santamarina, Wolfram Höps, Hufsah Ashraf, Nelson T. Chuang, Xiaofei Yang, Katherine M. Munson, Alexandra P. Lewis, Susan Fairley, Luke J. Tallon, Wayne E. Clarke, Anna O. Basile, Marta Byrska-Bishop, André Corvelo, Uday S. Evani, Tsung-Yu Lu, Mark J. P. Chaisson, Junjie Chen, Chong Li, Harrison Brand, Aaron M. Wenger, **Maryam Ghareghani**, William T. Harvey, Benjamin Raeder, Patrick Hasenfeld, Allison A. Regier, Haley J. Abel, Ira M. Hall, Paul Flicek, Oliver Stegle, Mark B. Gerstein, Jose M. C. Tubio, Zepeng Mu, Yang I. Li, Xinghua Shi, Alex R. Hastie, Kai Ye, Zechen Chong, Ashley D. Sanders, Michael C. Zody, Michael E. Talkowski, Ryan E. Mills, Scott E. Devine, Charles Lee, Jan O. Korbel, Tobias Marschall, Evan E. Eichler (2021). Science, 372(6537):eabf7117.

Chapter 2

Background

This chapter aims to introduce the preliminary concepts in biology, mathematics and computer science.

2.1 DNA and chromosomes

Deoxyribonucleic acid (DNA) is a hereditary molecule inside the bodies of living organisms that passes genetic information through generations. It has a *double helix* structure consisting of two *strands* bound together with hydrogen bonds. Each DNA strand is a *nucleic acid* molecule consisting of a chain of building blocks called *nucleotides*. Nuleotides are monomeric units of DNA with three parts: a nitrogen base, a five-carbon sugar and a phosphate group. There are four different types of nitrogen bases in a DNA molecule: *Adenine (A)*, *Cytosine (C)*, *Guanine (G)*, and *Thymine (T)*. The order of nucleotide bases defines a DNA strand, so DNA can be seen as a string over the alphabet {A, C, G, T}.

The hydrogen bonds binding the two DNA strands are only formed between specific pairs of bases that are complementary to one another. The complementary *base pairs*[1] are {A,T} and {C, G}, so knowing the sequence of one DNA strand uniquely determines the sequence of the complementary DNA strand. DNA strands have directions by which they are read in the biological machinery of cells. Each strand has two different ends called 5' and 3' and is always read from 5' to 3' end. The two strands of DNA, called *Watson* and *Crick*[2], are always in opposite directions, so these two strands are reverse and complement of each other.

The whole DNA content of an individual organism is called *genome*. DNA is packed in compact packages named *chromosomes* inside the cells. Each chromosome has a *centromere* with short (*p-arm*) and long (*q-arm*) parts of the DNA molecule at its two sides. Figure 2.1 shows a schematic of cell, DNA, and chromosomes. The human genome has 23 pairs of chromosomes including 22 pairs of *autosomes*[3] and one pair of sex chromosomes that determines the sex and is XX in females and XY in males.

The genome of nearly all mammals including human is *diploid*, which means that it

[1] base pair (bp) is used as a measure unit for the length of a DNA double strand.
[2] named after two scientists who discovered the DNA double helical structure
[3] non-sex chromosomes

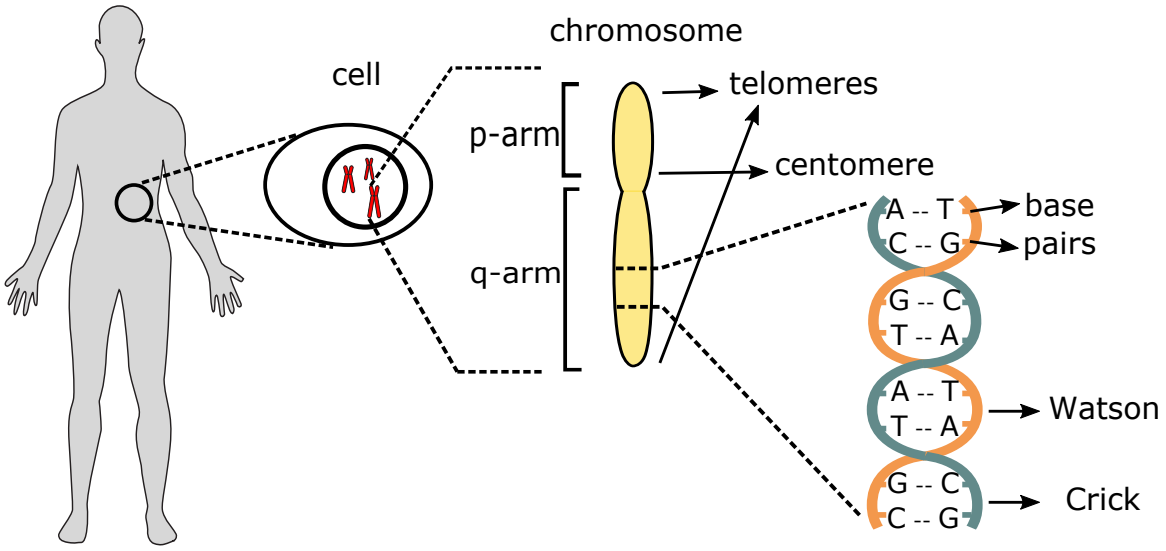


Figure 2.1: DNA and chromosomes. DNA is packaged into chromosomes inside the cells of living organisms. Each chromosome has two telomeres at its ends and one centromere in the middle part with p (short) and q (long) arms in the two sides of the centromere. DNA is a double helix molecule with two complementary strands called Watson (orange color) and Crick (teal color). Complementary base pairs are (A,T) and (C,G).

harbors two homologous copies of each chromosome. Each homolog is called a *haplotype* and is inherited from one parent. Some cells like human gametes[4] and bacteria cells are *haploid*, meaning that they have only one copy of each chromosome. There are also some organisms with *polyploid* cells, especially plants such as potato, that have more than two copies of homologous chromosomes. The number of homologous copies of each chromosome in a genome is called *ploidy*.

2.2 Genomic variants

Genomic variants contribute to the diversity in individuals' phenotypes[5] and a wide range of diseases, hence it is important to understand, explore and discover these variants. Variants are compared to a *reference genome*, which nowadays has been computed and been made available for many different species.

Genomic variants can be inherited from parents (*germline* variants), or be formed during an individual's lifetime (*somatic* variants). Since all cells are descendant from an initial fertilized egg, the initial DNA content (including germline variants) remains the same in the whole body of an individual. Somatic variants, in contrast, only exist in a sub-population of cells in the affected tissue.

There are three different types of variants with sizes ranging from a single nucleotide to millions of base pairs in the genome. We define and explain different types of genomic

[4] sperm and egg in human cells
[5] observable traits

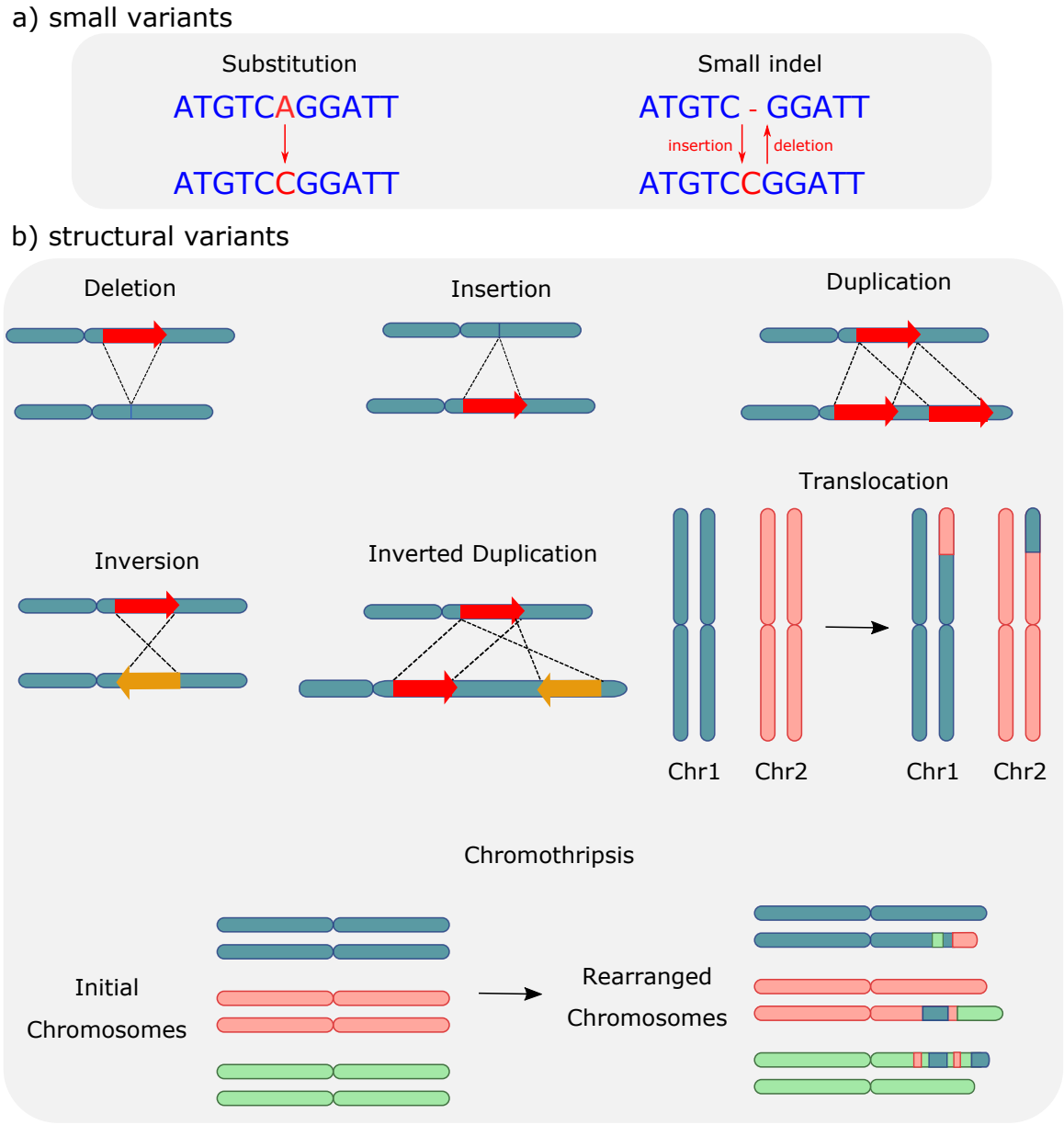


Figure 2.2: Genomic variants. a) Small scale variants including substitution and small indels. b) Structural variants including large scale copy number variants such as deletion, insertion, duplication, copy-neutral variants such as inversion, more complex variants such as inverted duplication, and chromosomal rearrangement events including translocation and chromothripsis.

variants with more emphasis on large-scale structural variants, which is mainly related to the next chapter in this thesis.

Figure 2.2 represents different types of variants in the genome. The most common type of variants are *Single Nucleotide Variants (SNVs)*, indicating substitution of one single base pair in the genome (Figure 2.2a, left). An estimated average number of SNVs per genome is reported to be between 4 to 5 million base pairs [16].

The second class of variants are short ($< 50bp$) *insertions* and *deletions (indels)*

(Figure 2.2a, right). Insertions (or deletions) result from adding (or deleting) one or a few base pairs to (from) a genome. Indels constitute between 3 to 5 million base pairs per individual genome [16].

Structural variants (SVs) (Figure 2.2b) are the other type of genomic variants that involve alterations of large genomic segments ($\geq 50bp$). Although SVs are less frequent than SNVs and small indels, they involve a larger part of the genome (between 10 to 12 million base pairs) [16], and have a large contribution to diversity of human phenotypes and disease [112].

Structural variants can not only change the dosage of genes, but also they can change the 3D structure of the genome and gene regulation mechanisms [102]. SVs have been shown to play key roles in many types of cancers [15, 72]. Moreover, SVs have been linked to many diseases and syndromes such as Crohn's disease, attention deficit hyperactivity disorder, diabetes, autism, schizophrenia, Down syndrome, Smith–Magenis syndrome (SMS), Potocki–Lupski syndrome (PLS) [112].

Structural variants can change and rearrange the structure of genome in many different ways. A set of different classes of common structural variants are presented below (see Figure 2.2b):

- *Copy number variants (CNVs)* are alterations of copy number of a genomic segment. They generally include classes of variants involving copy number changes such as insertion, deletion, and duplication.

- *Insertion* is a type of copy number variant in which a DNA segment is inserted into the genome.

- *Deletion* is a type of copy number variant in which a DNA segment is deleted from the genome.

- *Duplication* is an example of a CNV that happens when a segment is duplicated in the genome. The duplicated segments can be tandem (next to each other) or interspersed in the genome.

- *Inversion* is the replacement of a genomic segment with its reverse and complement sequence.

- *Inverted duplication* happens when a segment is inverted and duplicated at the same time. In this type of variant, the sequence of the duplicated segment is reverse and complement of the original segment.

- *Translocation* is the exchange of two ends of different chromosomes with each other.

- *Chromothripsis* is a massive shattering and rearrangement of genomic segments. It can occur in one or several chromosomes involving changing the order and orientation of genomic segments.

2.3 Genetic heterogeneity in cancer

Cancer is a genome disease created by accumulation of gradual or step-wise mutations resulting in a tumor tissue with heterogeneous clones and subclones. A *clone* is defined as the set of tumor cells with highly similar genotypes, and a *subclone* is a subpopulation of cells diverging in tumor evolution through acquiring new mutations. *Driver mutations* have fitness advantages in tumor evolution and can survive and create a tumor clone, while *passenger mutations* have no selective advantage and therefore no effect on the fitness.

A study on tumor evolution models [19] represents four different competing models: linear, branching, neutral, and punctuated. Different models can be seen at different time points of cancer evolution and in different classes of mutations or tumors. The presented models are based on the theory that cancer starts from a single cell inside a normal tissue.

The linear model suggests that mutations occur step by step, and a driver mutation can have a strong selective advantage, so that it takes over the whole tumor. This model was supported by a study on colorectal cancer [27] where the authors show that cancer progresses through a step-wise chain of mutations resulting in a more malignant stage of cancer.

In the branching evolution model, different clones can be created and diverge from a common ancestor. They can evolve at the same time leading to a tumor cell population with different clones. The branching evolution model has been supported in many cancer studies, such as breast cancer [78, 99, 109, 117], liver cancer [63], kidney cancer [35, 36], brain cancer [2, 45, 101], and many other cancer types.

The neutral evolution is similar to branching evolution with the difference that mutations do not have any selective advantage or fitness, and they can randomly become fixed or get lost in the population. In this model, there are many intermediate tumor clones leading to very high tumor heterogeneity. This model has shown to be consistent with one third of examined cancers collected from different cancer cohorts [115], meaning that the neutral evolution model can be supported by a remarkable fraction of tumors.

The punctuated evolution model suggests that a number of mutations and genomic aberrations occur at a very small burst of time at the early stage of cancer progression. This model has clones immediately branching at the early stages of tumor initiation, a few of which can survive and dominate the tumor mass. Unlike linear and branching evolution models, which are mainly supported by single-nucleotide mutations, the punctuated evolution model is usually supported by copy number changes, structural variations and chromothripsis. There have been studies reporting chromothripsis in several cancer tumors including colorectal cancer [49], prostate cancer [7], and ovarian cancer [83].

2.4 DNA sequencing

The problem of determining the order of nucleotide bases inside a DNA sequence is called *DNA sequencing*. The first fully sequenced genome was accomplished by Sanger

sequencing technology in 1977 [96]. At the end of the 20th century, the *next-generation sequencing (NGS)* technologies emerged, which were highly parallel, scalable and low-cost compared to the traditional Sanger sequencing. These technologies are based on *shotgun sequencing*, in which the DNA sequence is cut into random (relatively short) fragments that can be sequenced in parallel. The parts of the fragments that are read in a sequencer are called *sequencing reads*, which can be read either from one end (*single-read*) or from both ends (*paired-end*) of a fragment.

Second generation technologies such as *Illumina* [6] and *454 Pyrosequencing* [93] produce highly accurate reads of length 50bp up to 700bp depending on the specific technology. Third generation sequencing technologies, first described in 2009 [11], such as *PacBio* and *Nanopore* sequencing use different sequencing approaches that can produce reads longer than tens of thousands of base pairs, which are usually highly error prone. Recently, *HiFi* reads were developed by optimizing the *circular consensus sequencing (CCS)* method that is able to produce long accurate reads of up to 20000 base pairs [114].

Nowadays, it is possible to sequence the genome of individual single cells thanks to the recently developed single-cell sequencing technologies. Single-cell sequencing has been selected as method of the year 2013 by Nature Publishing Group [24]. Having a signature of genomic heterogeneity and clonal structure of cells, single-cell sequencing has become a unique technique to study evolutionary history of cells. These technologies are being widely used in evolutionary history of mutations among single cells, especially in cell populations with a high mutation rate such as cancer cells.

2.4.1 DNA assembly

The problem of reconstructing the genome using DNA sequencing reads is called *denovo genome assembly*. The assembly problem consists of two main computational steps:

- computing *unitigs* and *contigs* based on overlaps between reads, where a *contig* is a consensus sequence built from a set of overlapping reads, and a *unitig* is a sequence that can be unambiguously assembled from overlapping reads,
- *scaffolding* contigs into sets of ordered and oriented contigs; these ordered sets are called *scaffold*.

Genome assembly can be modeled by graphs representing the read overlaps in which the original genome (each haplotype) corresponds to a path. The commonly-used graphs in genome assembly are the *De Bruijn graph (DBG)*, which represents overlaps of length $k-1$ between *kmers* [6] [85], and the *overlap graph*, which is based on *overlap layout consensus (OLC)* paradigm and represents overlap between reads [73].

Figure 2.3 represents an example diploid genome with its De Bruijn and overlap graphs. If the subgraph induced from nodes x, y, z has the edges $(x, y), (y, z), (x, z)$, the edge (x, z) is called a *transitive edge* (gray dashed edge at Figure 2.3c). The *transitive edges* in the overlap graph usually get removed from the graph in assembly algorithms. Figure 2.3d shows the resulting simplified graph after merging simple linear

[6] all substrings of length k of a read

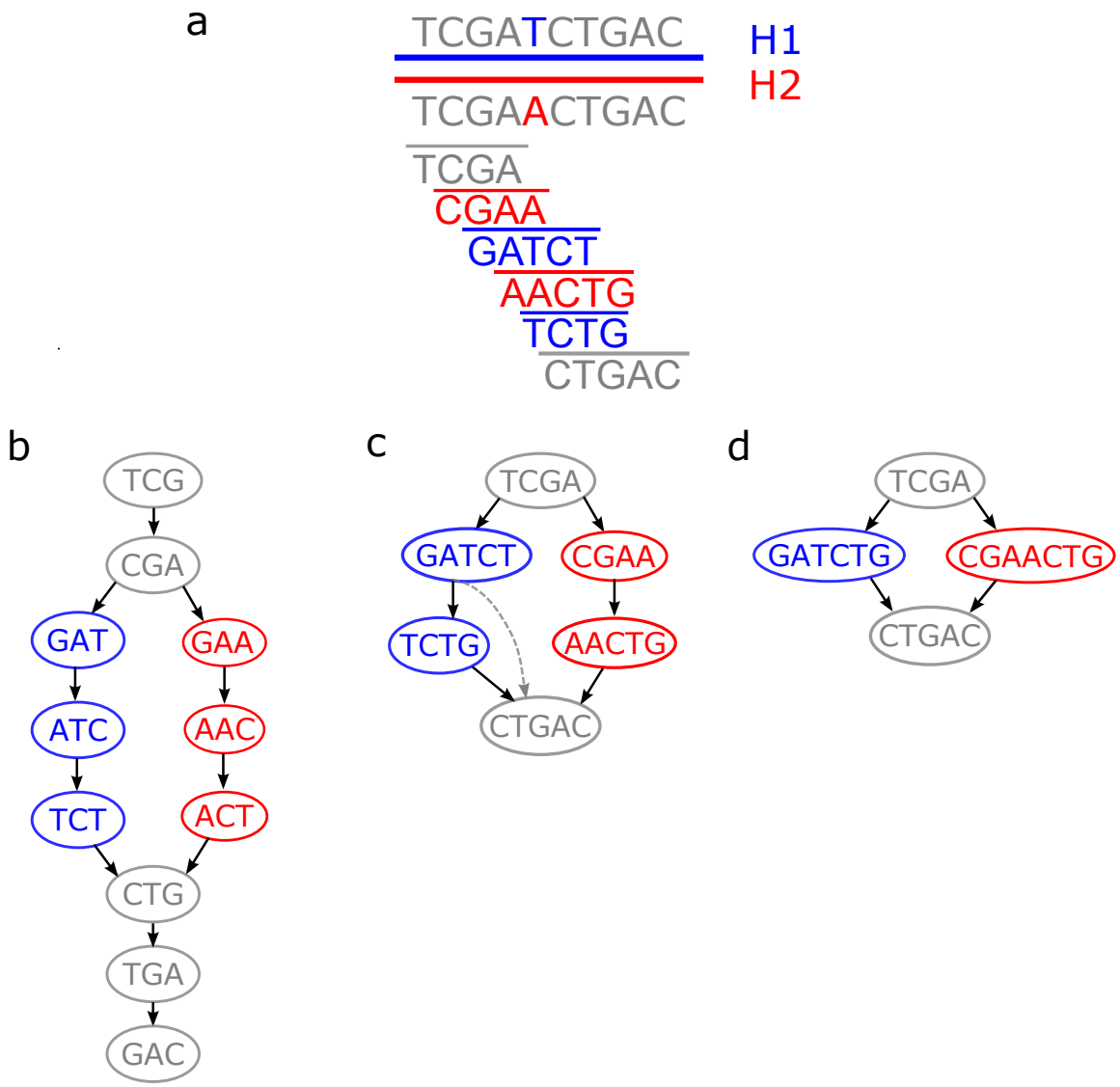


Figure 2.3: DNA assembly graphs. a) An example of a diploid genome with H1 (blue) and H2 (red) haplotypes including a heterozygous SNV. The sequencing reads are shown in gray, blue, and red colors showing homozygous reads and heterozygous reads from H1 and H2 haplotypes, respectively. b) A De Bruijn graph with kmer size equal to 3. Homozygous nodes are shown in gray color, and H1/H2 heterozygous nodes are represented in blue/red color. c) An overlap graph with nodes corresponding to sequencing reads and edges representing the overlaps of minimum length 2. The gray edge shown with dashed line is a transitive edge that gets removed from the graph in common assembly algorithms. d) A simplified graph after merging and collapsing simple paths into one node. The graph has bubble structure resulting from two divergent haplotype assembly paths.

paths in De Bruijn or overlap graphs. The structure of this compact graph is called *bubble* that occurs when there are two divergent paths of overlaps. A simple *bubble* is defined as an induced subgraph on a set of four vertices x, y, w, z with the edges

$(x, y), (y, z), (x, w), (w, z)$. Bubble structures are formed in genetic variants, which can represent variations between different sample genomes or heterozygous sequences in a diploid genome. They can be hence utilized to infer genomic variants or heterozygosity in *de novo* sequencing paradigms.

Since the emergence of NGS data, many assembly tools have been developed including Velvet [121], Celera [74], and SGA [100] for short read assembly and Falcon [14], Nanocorr [41], Canu [52], HiCanu [79], and Hifiasm [13] for long-read or hybrid [7] assembly.

2.4.2 Resequencing

Nowadays, a reference genome has been computed and made available in public biological datasets for many species. In the presence of a reference genome, we can find similar substrings in the reference genome for each read (*mapping*) to guess the original location of reads and then compute the differences between read sequences and the reference genome to find genomic variants (*variant calling*).

When a candidate mapping location in the reference genome is found for a read, it is usually needed to align the read to the corresponding reference substring. *Alignment* of two strings is defined as the process of transforming one string to another by mismatch or gap (insertion or deletion) operations. This transformation can be formalized as an optimization problem of maximizing the alignment score or minimizing the cost of mismatches and gaps.

Edit distance is an example of the alignment problem where the objective is to minimize the total number of mismatches and gaps, i.e., every mismatch or gap has a unit cost. Most of the alignment tools in genomic sequence alignment are based on affine gap cost where the cost of a gap of length L is $A + B \cdot L$ with parameters A and B indicating penalties for gap opening and extension, respectively.

There is a wide range of mapping and alignment tools in bioinformatics including bwa [59], minimap [57], and minimap2 [58]. These tools usually use heuristic algorithms to find a fast sub-optimal solution of the alignment problem.

After mapping and alignment, genomic variants can be called. There are distinct tools for different variant classes including GATK [21, 70] and FreeBayes [32] for calling SNV and short indels and Delly [89] for calling structural variants.

2.5 Genotypes and haplotypes

Different variations of a gene or genomic loci are called *alleles*. *Haplotype* is defined as a set of alleles coming from the same parent. In genomics studies, each haplotype is the DNA sequence of one parent in the genome, and the possible variants in a genomic locus[8] or region are called alleles. *Genotype* refers to the sequence of the set of paired haplotype alleles over all loci/regions in the genome. Figure 2.4 shows the difference between genotype and haplotype in a small example.

[7] using a combination of short and long reads

[8] location

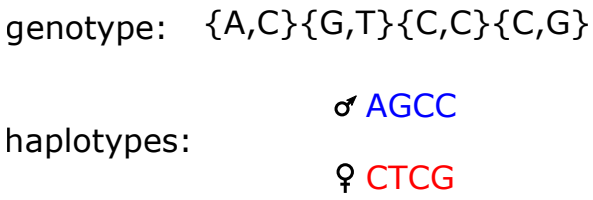


Figure 2.4: Genotype vs haplotype. A Genotype represents the genome of both (all) haplotypes at each locus. Haplotypes determine the parent alleles separately in the genome. In the given example, the third locus is homozygous, and the other loci are heterozygous.

If both (all) haplotypes at a genomic locus have the same allele, the locus is called *homozygous*, otherwise it is *heterozygous*. Assuming that heterozygous loci are all biallelic, a genome with n heterozygous loci can lead to 2^{n-1} potential haplotype configurations in a diploid genome. *Genotyping* is the problem of determining genotypes in the genome, and *haplotype phasing* refers to inferring the DNA sequence of haplotypes.

Haplotype phasing is important to study *compound heterozygosity* where the two haplotypes have mutations in two different genomic loci, which can result in malfunction of both copies of a gene. Moreover, knowledge of haplotypes can lead to a more accurate discovery of the relation between genome and phenotypes and differentiation of human populations [39, 105].

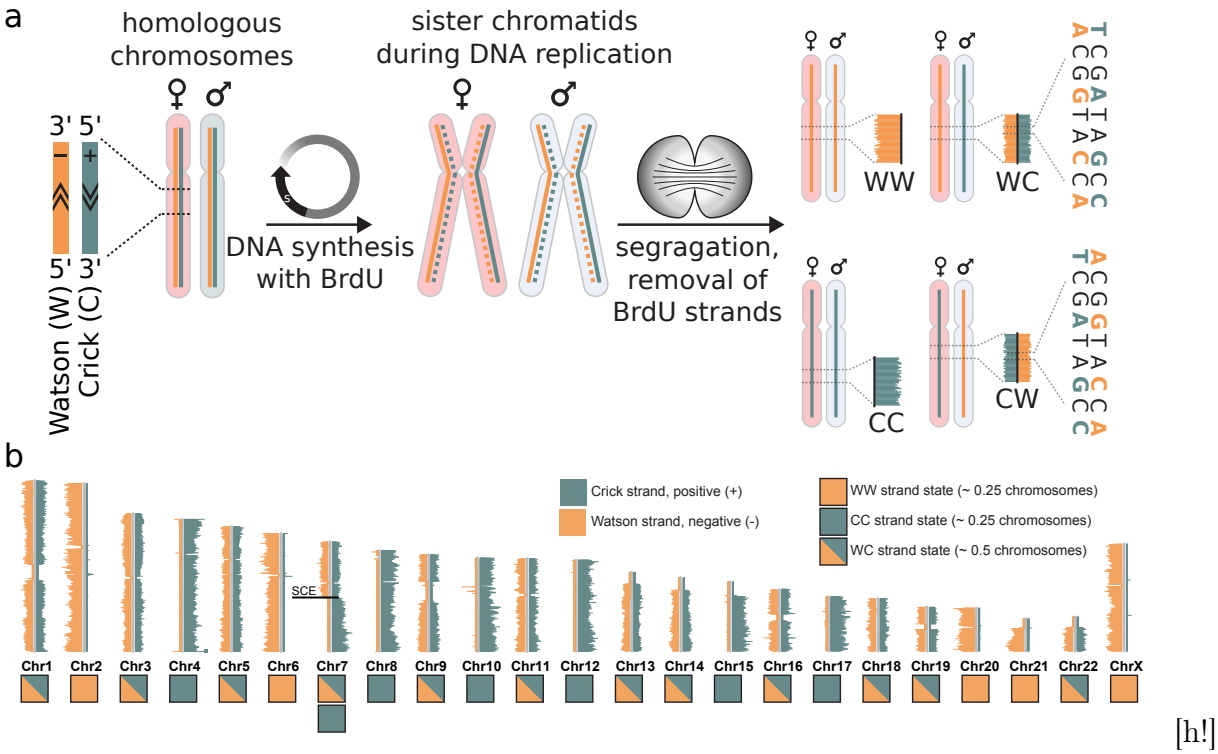
A common procedure of haplotype phasing using sequencing reads and a reference genome involves several steps including mapping reads to the reference genome, variant calling and genotyping, bi-partition of reads based on their corresponding haplotypes (referred to as *haplotagging* reads), and inferring haplotype alleles in heterozygous variant calls. There are various tools for reference-based haplotype phasing including WhatsHap [84] and HapCUT [5] and *de novo* diploid haplotype assemblers [14, 113].

2.6 Single-cell strand sequencing

Single-cell strand sequencing technology (Strand-seq) was developed in 2012 [26] and has been applied to study many biological problems such as inversion detection [94], haplotype phasing [86, 87], detecting structural variants [95], and chromosome clustering [37].

Figure 2.5-a shows the principle of Strand-seq technology. In a single cell division, *DNA replication* happens in which the original *template strands* of DNA detach from each other, and new strands form in the opposite directions.

These new strands are tagged with *bromodeoxyuridine (BRDU)* in the Strand-seq technique. The new DNA strands are then nicked at BRDU sites by UV light, which prevents them from amplifying in sequencing library preparation, so the sequencing reads are only sampled from the template strands. Four equally-likely possible *strand states* can happen for the daughter cell based on random segregation: *CC*, *CW*, *WC*, or *WW* depending on direction of the template strand (Crick (C) or Watson (W)) inherited from mother and father haplotypes. These random strand states happen independently for each single-cell and chromosome.



[h!]

Figure 2.5: Single-cell strand sequencing principle. a) Left: Two homologous chromosomes in a diploid cell each one with two DNA strands in Watson (orange) and Crick (teal) directions. Middle: DNA synthesis in presence of BrdU: formation of new (dotted) DNA strands tagged by BrdU after one round of cell division. Right: Removal of BrdU incorporated (dotted) strands result in four possible DNA strand states (WW, WC, CC, CW) in a daughter cell according to random segregation. WC and CW strand states are haplotype informative. b) Vertical ideograms show different chromosomes with their directional coverage from a single-cell Strand-seq library. Horizontal orange/teal lines show the coverage of Watson/Crick reads alongside the chromosomes. The example shows various inherited strand states in different chromosomes. Chromosomes with sister chromatid exchange (SCE) events (chromosome 7) show a combination of different strand states. Figure from [37, 86].

A key feature of Strand-seq data is that reads coming from a Watson or Crick strand map in forward or reverse direction to a reference genome, respectively. The cells of type WC or CW in a chromosome imply a direct separation of reads by their haplotypes (Figure 2.5a,right) because the reads that map in opposite directions come from different haplotypes. In the remainder of this thesis, the cells are assumed to be diploid unless it is mentioned otherwise, and when the haplotype is not known, we use WC to refer to both WC and CW strand states since these two classes are not distinguishable before haplotype phasing.

Due to independent random segregation, we observe different strand states in chromosomes for a single-cell (Figure 2.5b). *Sister chromatid exchanges (SCE)* are the only exception that create a switch in the strand state alongside a chromosome (chromosome 7 in Figure 2.5b). SCEs are the exchanges of genomic segments between two

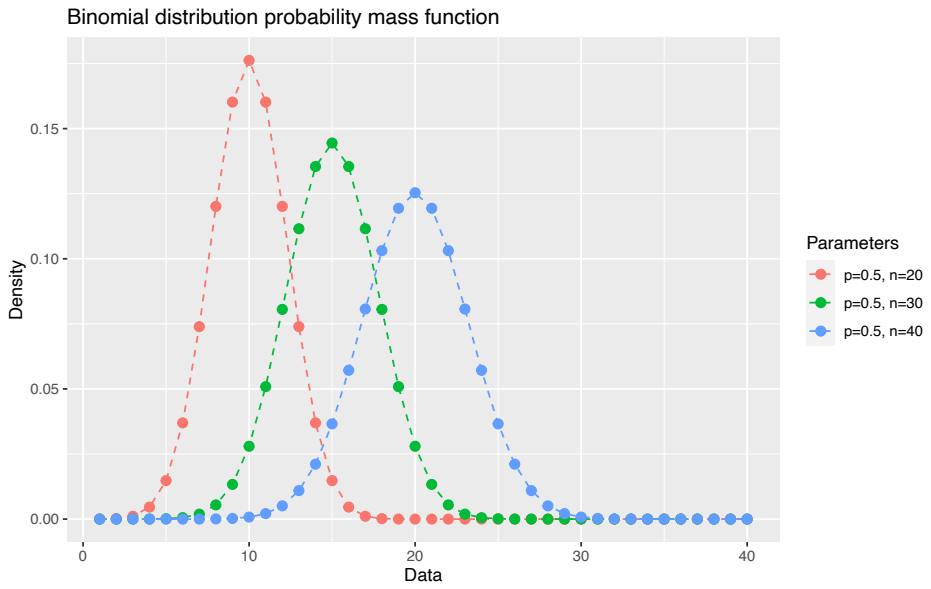


Figure 2.6: Binomial distribution. Binomial probability mass functions for three different binomial distributions with $p = 0.5$ and different n parameters $20, 30, 40$ shown in blue, green, and red colors, respectively.

identical chromatids. They are a source of genome instabilities created in repairing mechanisms for double strand breaks [40]. Detecting SCEs in single-cells is the first biological problem solved by Strand-seq technology[26].

2.7 Probability and statistics background

This section introduces the required concepts in probability and statistics.

2.7.1 Binomial distribution

The Binomial distribution models the number of successes in a set of independent repeated experiments with the same success probability. This distribution has two parameters: p denoting the success probability and n denoting the number of repeated experiments. The probability mass function, mean, and variance for a binomial random variable X with parameters p and n are as follows:

$$\begin{aligned} Pr(X = k) &= \binom{n}{k} p^k (1-p)^{n-k} \\ E(X) &= np \\ \sigma^2(X) &= np(1-p) \end{aligned} \tag{2.1}$$

Figure 2.6 shows examples of three binomial probability mass functions with $p = 0.5$ and different n parameters $20, 30, 40$.

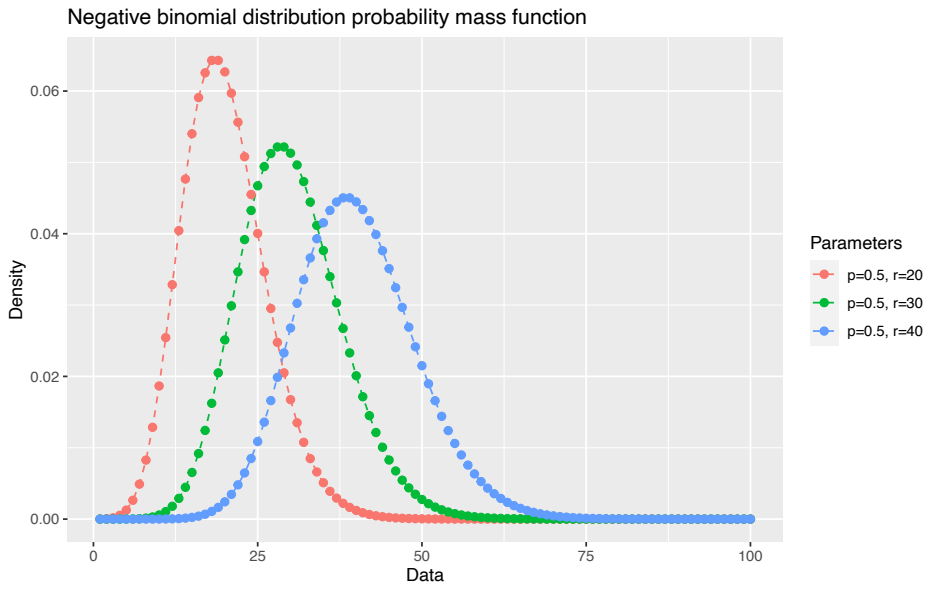


Figure 2.7: Negative binomial distribution. Negative binomial (NB) probability mass functions for $p = 0.5$ and three different dispersion parameters $20, 30, 40$ shown in blue, red, and green colors, respectively.

2.7.2 Negative binomial distribution

Negative binomial (NB) distributions are a generalization of the Binomial distribution with a parameter p and an extra dispersion (r) parameter. It models the number of successes in a set of independently and identically (iid) distributed Bernoulli trials until a specific number of failures (r) happens. An NB random variable X with parameters p and r has the following probability mass function, expectation, and variance:

$$\begin{aligned} Pr(X = k) &= \binom{k + r - 1}{r - 1}(1 - p)^k p^r \\ E(X) &= \frac{pr}{1 - p} \\ \sigma^2(X) &= \frac{pr}{(1 - p)^2} \end{aligned} \tag{2.2}$$

Figure 2.7 shows the probability mass functions for three different NB distributions with $p = 0.5$ and different dispersion parameters $20, 30, 40$. As shown in the figure, dispersion and variance of the distribution increases with higher r values.

2.7.3 Modeling genomic read counts

Analysis of read counts in genomic regions is a fundamental task in various genomic studies including differential gene expression analysis and copy number variant detection. In a simplified scenario where the coverage of sequencing reads is uniform all over the genome, the number of sequencing reads in a fixed-sized genomic interval fit to the binomial distribution because different regions of the same size have the same proba-

bility of being the original interval where a read is sampled from. However, sequencing technologies have systematic biases in read coverage that lead to the over-dispersion problem where the read count distribution of different fixed-sized bins have higher variance (dispersion) compared to the uniform-coverage scenario. The NB distribution has been proposed to model genomic read counts because it captured the over-dispersion problem arising from non-uniform sequencing read coverage [64]. This model fits well to the number of reads in genomic regions and is commonly used for modeling read counts.

2.7.4 Mixture models and EM algorithm

Mixture models are mixtures of probability distributions used in statistics to model observed data coming from different subpopulations of an overall population. The subpopulation or component memberships for different data observations are not known in mixture models, and they are viewed hidden random variables that should be inferred using statistical methods such as maximum likelihood estimation.

The assumption in a mixture model is that different subpopulations have the same distribution with different parameters. More precisely, a mixture model with K subpopulations can be defined with the following components:

- A discrete probability distribution representing the prior probabilities of subpopulation memberships. It has a set of k mixture weights summing up to 1.
- A set of k parameter settings for the k different subpopulation probability distributions.

Figure 2.8 shows a mixture of two normal distributions with equal mixture weights $(0.5, 0.5)$ and parameters $\mu = -2, 2$ and $\sigma^2 = 1$. As shown in the figure, the mixture density function is the normalized sum of the two normal density functions. In the case of non-uniform mixture weights, the mixture distribution would be proportional to the weighted sum of distributions.

Parameter estimation and component membership inference

The parameters of a mixture model including mixture weights and different distribution parameters can be hidden as well as the component membership information of the observed data. *Decomposition* of a mixture model is defined as the problem of inferring the probability distributions and parameters that constitute the mixture model. Several approaches have been proposed to solve the decomposition problem of mixture models, many of which are based on maximum likelihood (ML) or maximum a posteriori (MAP) estimation methods.

Expectation Maximization (EM) algorithm is a popular method to estimate the mixture model parameters. It can be used as a soft clustering algorithm for computing the mixture components membership probabilities for different observed data.

EM is an iterative numerical algorithm for solving the maximum likelihood (ML) or maximum a posteriori (MAP) problem when there is no closed-form solution for these optimization problems [20]. It has been proved to converge to a local maximum in the aforementioned problems and has been widely used in statistical inference.

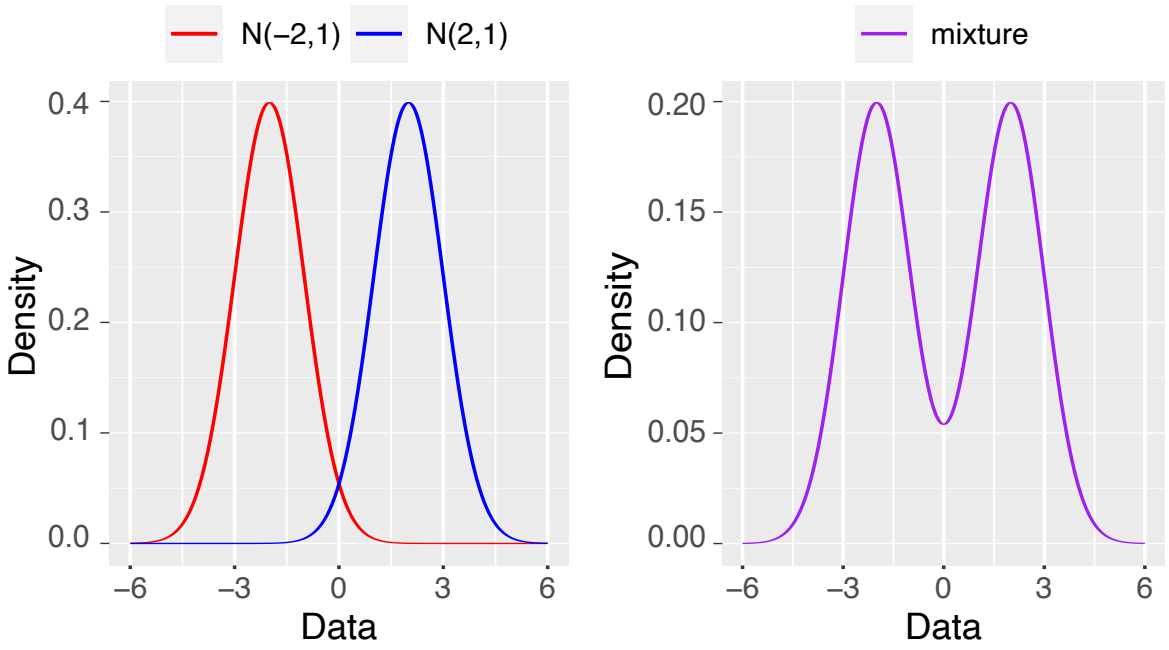


Figure 2.8: Mixture model. a) Probability density functions for two normal distributions with parameters $(\mu = -2, \sigma^2 = 1)$ and $(\mu = 2, \sigma^2 = 1)$ represented with red and blue curves, respectively. b) Probability density function of the mixture of the two normal distributions. The mixture weights are equal to $(0.5, 0.5)$, and the density function of the mixture distribution equals to the normalized sum of the two normal density functions.

2.7.5 Probabilistic graphical models and plate notation

A *probabilistic graphical model (PGM)* is a graph model that represents dependencies between random variables in a probabilistic model. PGMs are commonly used in probability, statistics, and machine learning.

There can be different types of directed and undirected graphical models. In this thesis, we use *Bayesian networks* that are directed acyclic graphical models showing conditional dependencies between variables. In probability theory, *conditional dependence* between events A and B happens if A and B are independent of each other, but they become dependent conditioned on occurrence of a third event C. It can happen if both A and B affect the probability of event C. For example, the events of having a rainy day and a windy day in the upcoming week are independent of each other, but given the information that the weather will be polluted, the probability of both events decrease and become dependent on each other. Conditional dependence can be defined accordingly in random variables if their related events are conditionally dependent on each other. In Bayesian networks, conditional dependence of random variables X and Y given Z is shown as a graph of three vertices X, Y, Z and two edges $(X, Z), (Y, Z)$.

A PGM can potentially have repeated components that can happen for example because of drawing multiple samples from the same distribution. The repetitions in PGM representation can be avoided by a compact version of graphical models, named *plate notations*. Plate notations represent each repeated component of a PGM by a

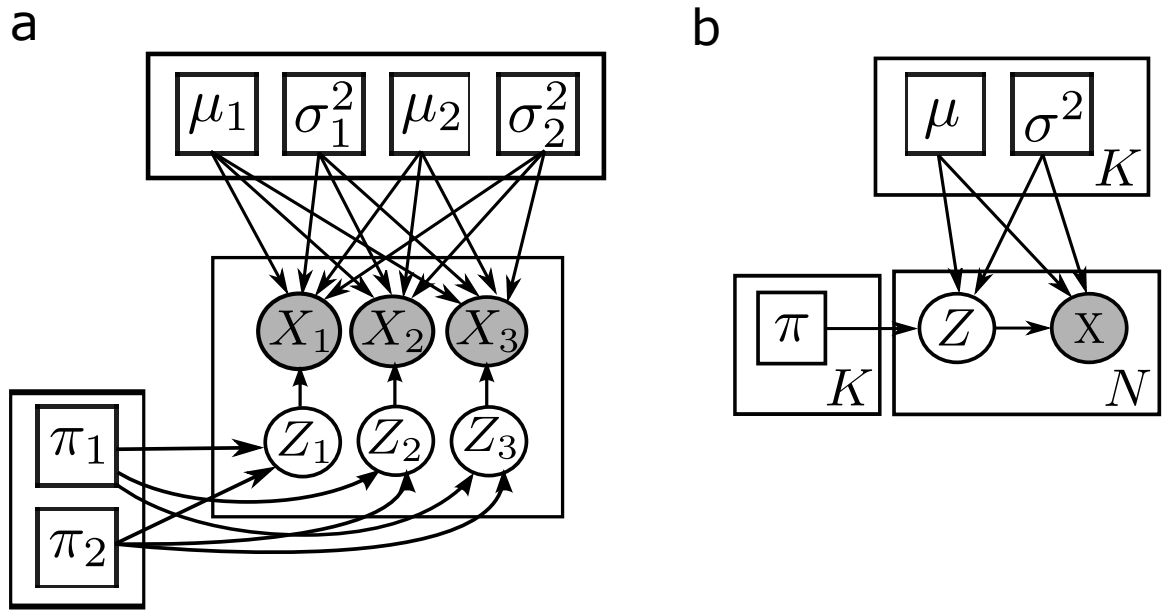


Figure 2.9: Probabilistic graphical model (PGM). a) Probabilistic graphical model (PGM) for the mixture of two normal distributions $\mathcal{N}_1(\mu_1, \sigma_1^2)$ and $\mathcal{N}_2(\mu_2, \sigma_2^2)$. Rectangles represent parameters, and circles represent random variables. The observed components are filled with gray color. There were three sample observations X_1, X_2, and X_3 in this example. The membership of each of these observations to $\mathcal{N}_1$ or $\mathcal{N}_2$ are hidden random variables Z_1, Z_2, and Z_3, which are generated from discrete distribution of mixture weights $\pi = (\pi_1, \pi_2)$. b) Plate notation of the same probabilistic model representing a compact form of PGM. Repeated components of PGM are shown by big rectangles (plates) with the repetition numbers on their bottom right corner. In this example, the repetition numbers are $k = 2$ and $N = 3$.

large rectangle (plate) with the repetition number written at the bottom right corner of the plate. Figure 4.2 shows a Bayesian network and a plate notation of a mixture model of two normal distributions (see Section 2.7.4).

Chapter 3

scTRIP: single-cell SV detection with tri-channel processing

This chapter represents a computational workflow for detecting structural variants in single cells. In this project, I shared first authorship (perceived as equal contribution) with Ashley Sanders, Sascha Meiers, and David Porubsky, and the project was done under supervision of Tobias Marschall and Jan Korbel. The results of this study were published in Nature Biotechnology [95].

In this project, I was involved in developing a data analysis workflow for single-cell SV detection together with Sascha Meiers, David Porubsky, Tobias Rausch, and Tobias Marschall. My main contribution was development of theory and implementation of a Bayesian statistical model for SV classification, which was one of the core parts in the SV discovery pipeline. Development of a multinomial distribution for haplotype-specific SV discovery was also done by me. I also contributed in analyzing single-cell clustering in Breakage-Fusion-Bridge cycle (BFB) events together with David Porubsky. I was mainly in charge of computing copy number confidence intervals for single cells in BFB cycles.

The other parts of this project were done by other authors, including single-cell Strand-seq library preparation, generation of single-cell data for the patient samples, cell mixing experiments, SCE detection tool, segmentation algorithm, bulk sequencing and gene expression data analysis.

The manuscript of the paper has been written in different parts by different authors. I specially contributed in creating the scTRIP workflow figure, and I was in charge of writing the methods and creating the probabilistic graphical model figure for the Bayesian statistics analysis for SV classification. The content and figures of this chapter is reused from the paper manuscript, with edits and adjustments for this thesis and more emphasis on the part of the project to which I contributed.

3.1 Introduction

Subclonal cell expansion is driven by mutation and selection, which can lead to cancer. In many cancer types, SVs serve as the leading class of somatic driver mutations [15, 72]. Structural variants can be either inherited through the germline and be clonal, or can

be formed *de novo* in somatic cells, defined as *somatic SVs*. Discovery of somatic SVs in single cells is essential to elucidate clonal evolution in cancer tissues [29, 104].

Discovering somatic SVs, especially copy-number-neutral and complex variants, in single cells has been challenging. Current reference-based SV discovery methods are based on signatures of discordant paired-end or split-reads that traverse breakpoints [51] or read-depth analyses, which is only suited for detecting CNVs [1, 29]. Since bulk sequencing data is a mixture of all different subclones in a cell population, detecting subclones of complex SVs such as somatic translocations, inversions and complex DNA rearrangements needs deconvolution, which is a very challenging problem. Single-cell sequencing is well-suited to detect subclonal variants and overcome these limitations [77], however the current single-cell SV discovery methods are limited to detect somatic CNVs [4, 34, 119].

We developed single-cell tri-channel processing (scTRIP), a computational framework to discover SVs (including somatic SVs) in single cells. scTRIP integrates read depth, strand and haplotype phase information from Strand-seq data to detect single-cell SVs. Strand-seq has been successfully applied in detecting germline inversions [94]. Tools for detecting diverse classes of structural variants in single cells were still missing prior to development of our framework. scTRIP can discover a wide variety of haplotype-resolved SV classes in single cells, facilitating studies of clonal evolution and SV formation mechanisms, which could lead to improvements in disease classification for precision medicine.

3.2 Structural variants signatures in Strand-seq data

There are specific footprints of Structural Variants (SVs), including inversions, in Strand-seq data. The strand-specific signal, together with read depth and haplotype information make Strand-seq technology a unique technique for detecting a wide range of SVs in single cells. In this chapter, we elaborate on haplotype-specific SV signals with examples of different SV types.

Strand-seq has random strand states for each single cell and chromosome. In addition, WC single cells are haplotype informative, that is, Watson and Crick reads covering a heterozygous SNV can be tagged by their original haplotype. This property leads to three layers of information for SV discovery (Figure 3.1.a): 1) **coverage layer**: the number of reads mapped to a region. 2) **strand layer:** the proportion of reads mapped in W and C directions, and 3) **phase layer:** the number of W and C reads tagged with H1 or H2 haplotypes. The red and blue circles in Figure 3.1a show H1 and H2 haplotype alleles in heterozygous SNV loci. In single cells of type WC, Strand-seq reads overlapping with heterozygous SNV positions can be tagged by their corresponding haplotypes (as shown in Figure 3.1 with red and blue lollipops for H1 and H2 alleles). These three layers of information in Strand-seq data enable us to detect a wide range of SV classes.

Figure 3.1 shows these three channels for normal reference state and different classes of structural variants. Note that all of the examples in the figure show heterozygous SVs in H2 in a single cell with WC state, i.e., W state in mother (H1) haplotype and C state in father (H2) haplotype. In a genomic segment with no SV (reference

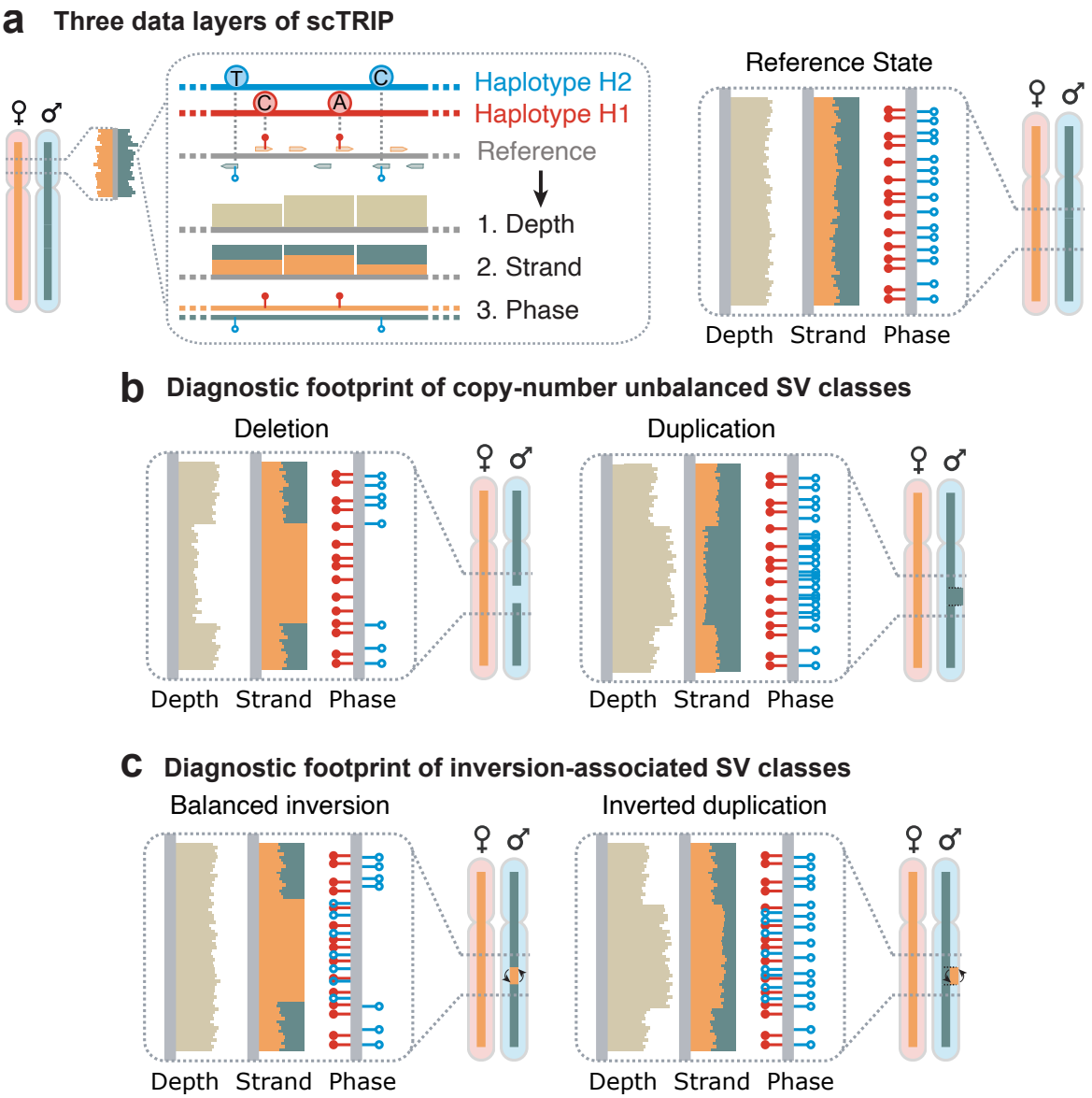


Figure 3.1: SV signatures in Strand-seq data. a) Left: An example of a single cell of type WC. W (orange color) is from mother haplotype (H1) and C (teal color) is from father haplotype (H2). Three data layers of information exist in Strand-seq data: (1) depth, (2) strand, (3) phase. b) Signatures of copy-number changes: Left: deletion with drop in coverage, change in the proportion of W reads (f_w.) from 0.5 to 1, and loss of H2 reads. Right: duplication with rise in coverage, change in f_w. from 0.5 to 0.33, and increased density of H2 reads. c) Signatures of inversions: Left: inversion with no change in coverage, change in f_w. from 0.5 to 1, loss of H2 reads in C direction and mixture of H1 and H2 reads in W directions. Right: inverted duplication with rise in coverage, change in f_w. from 0.5 to 0.66, and mixture of H1 and H2 reads in W direction.

state) (Figure 3.1a, right), we have a uniform read coverage in the depth channel, equal proportions of Watson and Crick reads in the strand channel, and clear separation of H1 (red lollipops) and H2 (blue lollipops) reads by their mapping direction in the phase channel (Watson and Crick haplotagged reads are shown on the left and right side of the bar in the phase channel, respectively).

In case of deletion of a segment in H2 (Figure 3.1b, left), we observe a drop in coverage (depth channel), a switch in Watson reads fraction *(f_w)* from 50% to 100% (strand channel), and loss of reads tagged with H2 (phase channel). In duplication (Figure 3.1b, right), we have a rise in coverage, switch of f_w to 33%, and double density of reads tagged with H2.

Strand-seq data also has a clear signature for inversions and other copy-number neutral (balanced) SVs. In a balanced inversion in H2 (Figure 3.1c, left), the coverage signal does not change, but Strand-seq specific signatures including strand and phase data are informative. There is a switch in f_w to 100%, and all phased reads (from H1 and H2) are in Watson direction in the inverted segment. More complex events like inverted duplications are also visible in Strand-seq data (Figure 3.1c, right). In this case, there is an increase in the coverage, alteration of f_w to 66%, and a mixture of H1 and H2 reads in the Watson direction. The phase channel makes this SV class distinguishable from duplication.

3.3 scTRIP computational pipeline

We developed a single-cell SV calling framework based on the aforementioned principles of Strand-seq data. Our pipeline is able to detect duplication, deletion, inversion, inverted duplication, and other types of more complex structural variants in single cells. Figure 3.2 shows an overview of the scTRIP pipeline. The details of each step in the pipeline will be explained in the following sections, with more emphasis on the parts that I have contributed and a more brief description on the other parts of the pipeline.

The scTRIP framework aligns, normalizes and counts W and C reads in fixed-size bins for each single cell (Figure 3.2a, left). It then performs a joint segmentation algorithm to fit piecewise constant functions to the bin W/C read counts with minimum sum of squared error (Figure 3.2b). The segmentation is done over all single cells and finds a set of potential SV breakpoints. Afterwards, the StrandPhaseR algorithm [87] is used to build consensus haplotypes from all WC cells using the set of heterozygous SNV positions, which results in chromosome-length haplotypes (Figure 3.2c). The consensus haplotypes are then used to haplotag Strand-seq reads covering the heterozygous SNV positions.

Given the strand states, segmentation and phase information, a Bayesian model is employed to compute posterior probabilities of different SV classes (Figure 3.2d). Every SV state implies a certain W and C copy number in each single cell based on the *ground strand state* of a segment. The *ground strand state* of a single cell is defined for a whole chromosome or a chromosomal region as the strand state that the single cell is expected to have in the absence of structural variants in the corresponding part of the chromosome. The Bayesian model consists of two parts. The first part is a Negative

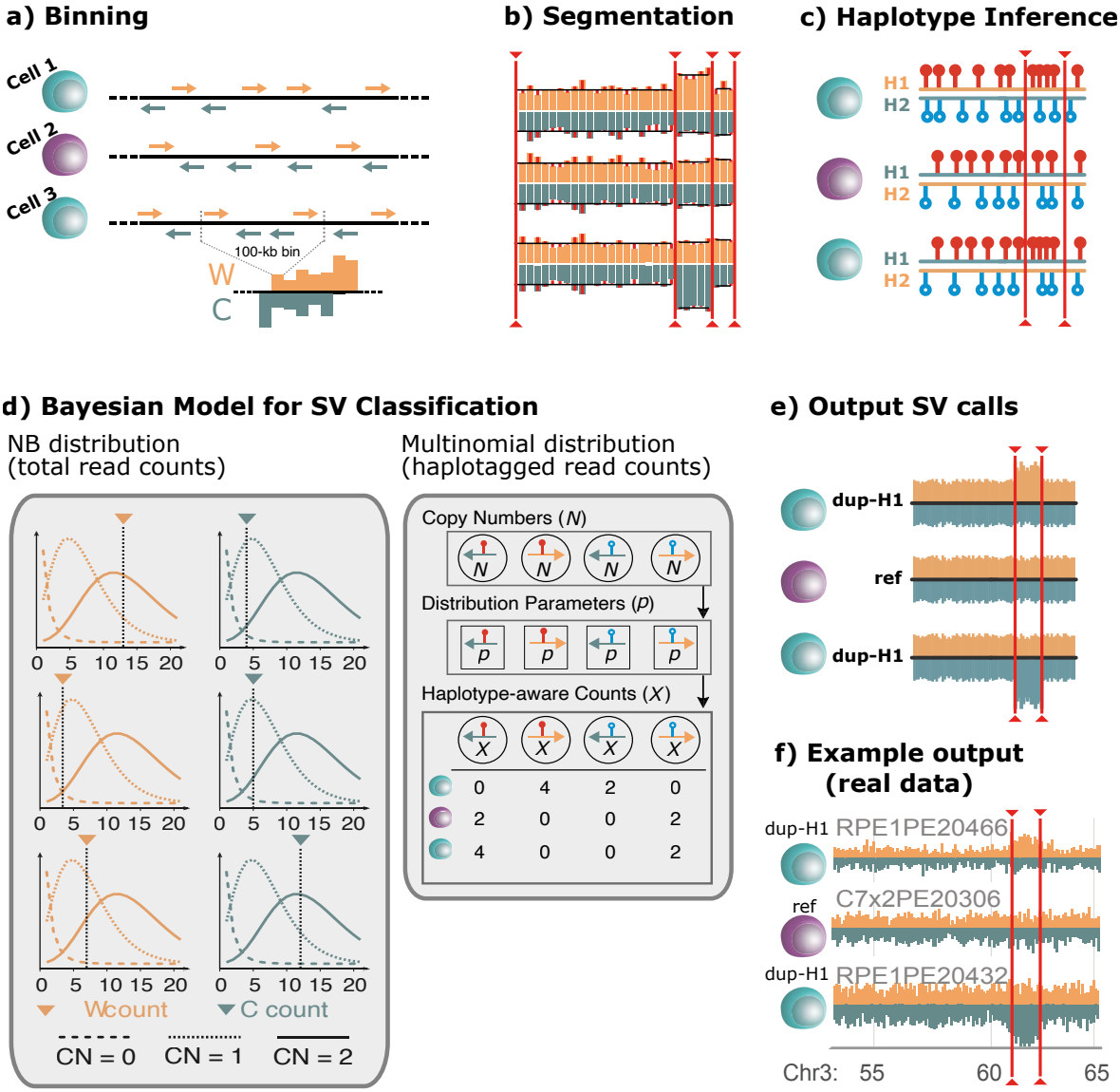


Figure 3.2: scTRIP workflow. a) Aligning, normalizing and counting W (orange) and C (teal) reads in fixed-size bins. b) Joint segmentation algorithm for locating the set of potential SV breakpoints. c) Building consensus chromosome-wide haplotypes based on WC cells in heterozygous SNP loci. d) Bayesian statistical model for SVs in single cells: Left: Negative Binomial (NB) distribution for the total W/C read counts. The likelihoods of the observed W/C read counts (vertical lines) can be computed from the implied W/C copy numbers from the SV states. Right: Multinomial distribution for the haplotagged read counts. Each SV gives rise to a certain copy number (N circles) for these four classes: C from H1, W from H1, C from H2, and W from H2. The p parameters (p rectangles) are proportional to these copy numbers generating the observed read counts (X circles) in the four classes. e) A schematic output of the scTRIP pipeline. There are two cells with duplication in H1 (cells with green color) and one cell with no SV (cells with pink color). d) An example SV calling output of scTRIP on real data (mixture of RPE-Wt and RPE-C7 cell lines) on chromosome 3.

Binomial (NB) distribution for the total number of W and C reads in segments/cells. For each single cell and segment, the likelihood of an SV state can be computed based on the W and C copy numbers implied from the SV state. The final output of the scTRIP is the set of most probable SV calls for each segment and cell. The second part is a multinomial distribution for modelling the number of W and C haplotagged read counts. Every Strand-seq read falls into one of the four categories: C read from H1, W read from H1, C read from H2, and W read from H2. An underlying SV class leads to a certain copy number (N circles in the figure) for each of these four classes for a single cell, which in turn implies p parameters (p rectangles in the figure) of the multinomial distribution. This multinomial model generates a certain number of reads in each of these four classes.

The final output of the scTRIP pipeline is a set of SV calls per genomic segments in single cells (Figure 3.2e). An example output of an SV call on real data from RPE-Wt and RPE-C7 cell lines is shown in Figure 3.2f.

3.3.1 Input data

The input data of scTRIP consists of a set of Strand-seq single-cell BAM files from the same donor sample. BAM files are conventional files (*.bam) that are a compressed binary version of SAM files (*.sam) storing the genome alignments. We use GRCh38 as human reference genome in our study. An optional input of scTRIP is a set of SNPs for locating heterozygous loci and haplotype phasing. In the absence of input SNP data, scTRIP can directly call SNPs from the Strand-seq single-cell data. We use a set of SNPs provided by the 1000 Genomes Project (1000GP; phase 3) in our study. Moreover, scTRIP requires an input tab-separated file with normalization factors (see below) per bin across the genome.

3.3.2 Binning single-cell read counts

Each chromosome is partitioned into fixed-size bins with the default length of 100kb. scTRIP counts the number of Watson and Crick reads in each bin per single cell. It excludes non-primary and supplementary alignments, PCR duplicates, and reads with low mapping quality (less than 10). The second reads of read pairs are also excluded in case of paired-end data in order to avoid double counting.

3.3.3 Coverage normalization

Sampled sequencing reads are known to have systematic coverage biases that can be related to GC content, local DNA structures, DNA fragmentation, or different mappability [25]. scTRIP normalizes single-cell read coverage to adjust for systematic coverage fluctuations. It uses Strand-seq data of 1058 single cells from the Human Genome Structural Variation Consortium (HGSVC-1) to estimate normalization factors. This data consists of 9 cell lines from the 1000 Genomes Project including the following samples: NA19238, NA19239, NA19240, HG00731, HG00732, HG00733, HG00512, HG00513, and HG00514 [10]. Based on a linear model, the pipeline derives normal-

ization scaling factors. It uses these scaling factors to rescale the W/C read counts in bins for all single cells prior to SV calling.

3.3.4 Segmentation

For segmentation, scTRIP uses the tiling array algorithm [46] with the minimum sum of squared error (SSE) objective function. Given an input parameter k as the number of breakpoints, this method is based on a dynamic programming algorithm to locate k optimal changepoints with minimal sum of squared error (SSE) [46]. The cost function of the algorithm is adjusted in scTRIP to count bin read counts in each cell and strand separately. The number of breakpoints is chosen as the smallest number k such that $SSE_k - SSE_{k+1}$ is smaller than a user-defined threshold (default: 0.1 used in this study).

3.3.5 Detecting SCEs and strand states in single cells

Signatures of a structural variant depend on the underlying strand state in each single cell and chromosome. These ground strand states (CC, WC, or WW) are constant all over each chromosome in every single cell unless there is an SCE (sister chromatid exchange) event [26]. An SCE results in a change in strand state of a single cell along the chromosome that is detectable and distinguishable from SV changepoints in scTRIP. An example SCE event in Strand-seq data is shown in Figure 2.5b, where an SCE exists in chromosome 7 and changes the strand state from WW to WC.

Unlike SVs, SCEs happen independently in each single cell, hence it is very unlikely to observe recurrence of SCE breakpoints at the same locus in more than one single cell in a sample. Changepoint recurrence is a key criterion in scTRIP to distinguish SCE and SV breakpoints.

To locate SCE breakpoints, scTRIP performs the same segmentation algorithm for each single cell separately. The ground strand state for each resulting segment between two breakpoints is assigned to WW if $f_w = \frac{W}{W+C} > 0.8$, to CC if $f_w < 0.2$, and to WC/CW otherwise. Note that we do not distinguish between WC and CW at this step. These two strand states will be distinguished using StrandPhaseR [87] in the haplotype phasing step. The pipeline then locates SCEs as single-cell segmentation breakpoints where a strand state changes that does not coincide with (more than 500kb far away from) a joint segmentation (SV) breakpoint. More precisely, all newly detected breakpoints at this step that have not been detected previously as SV breakpoints are selected as SCE breakpoints.

3.3.6 Haplotype phasing

In order to have haplotype-aware SV calling, our pipeline phases all chromosomes using the StrandPhaseR tool [87]. For each single cell, StrandPhaseR assigns regions with WC/CW strand states as either WC or CW. It is a greedy algorithm that uses the set of covered heterozygous SNVs shared between single cells to determine if W reads represent H1 and C reads represent H2 (denoted as WC), or vice versa (denoted as

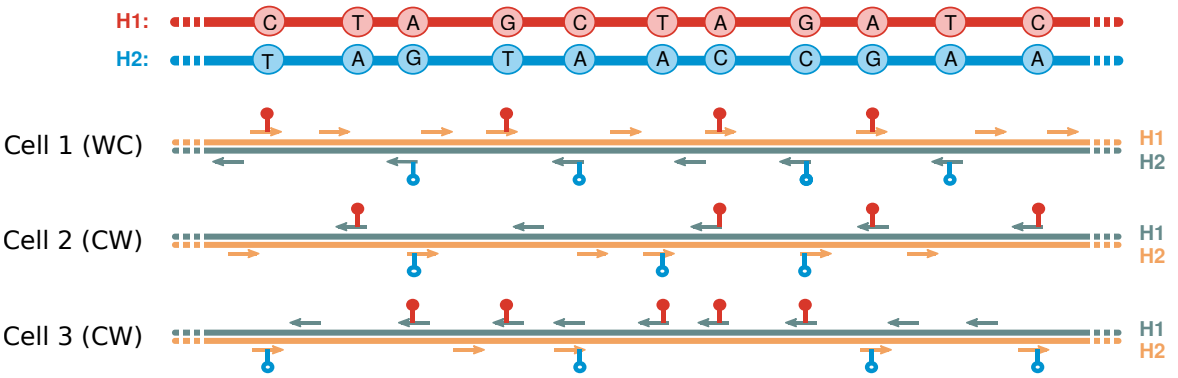


Figure 3.3: Strand-seq phasing signal Strand-seq cells of type 'WC' are haplotype informative in each chromosome. The (H1/H2) haplotypes can be assigned to W/C strand states based on the set of shared heterozygous SNV loci. H1/H2 reference genomes, SNV alleles, and tagged reads are shown in red/blue colors. The haplotype-specific strand states (WC, CW, CW for cell1, cell2, cell3) are computed in StrandPhaser greedy algorithm.

CW). Figure 3.3 shows the haplotype signal and the idea of StrandPhaser in an example of three Strand-seq single cells.

StrandPhaseR creates two matrices for W and C reads covering heterozygous SNV sites. Single cells correspond to rows and heterozgous SNV loci correspond to columns in each of these matrices. If an SNV allele is covered by a read from a single cell, the allele is written in the W or C matrix (depending on the direction of the read) in the corresponding row and column. The problem of haplotype phasing of single cells with WC strand state in a chromosome can be solved by assignment of each row in the aforementioned W and C matrices to H1 and H2, or vice versa. The main idea behind these haplotype assignments is that the covered alleles inside one haplotype should not conflict with each other over different single cells.

StrandPhaseR solves this problem by iteratively swapping equivalent rows (rows corresponding to the same single cell) between the two W and C matrices to minimize the number of haplotype allele conflicts. Note that exchanging a single cell's rows between the two matrices is equivalent to swapping haplotype assignment to W and C directions in that single cell.

StrandPhaseR first sorts single cells in matrices in decreasing order based on their number of covered SNV loci. It then computes the matrix conflict scores as total number of conflicts or disagreements over all SNV columns in both matrices, which show the total number of disagreeing SNV alleles in the initial haplotype assignment configuration. It then swaps the first row of the two matrices and recomputes the conflict score. If the swap operation improves the conflict score, it approves the swap; otherwise it keeps the previous configuration. The procedure is repeated for all single-cell rows to sort W and C direction of single cells in order to minimize the conflict in haplotype assignment. It can be repeatedly done in several iterations until the conflict scores do not improve anymore. In the original paper, two iterations over all matrix rows have been observed to be sufficient on the applied datasets.

The resulting optimized matrices can be used to compute the consensus haplotypes,

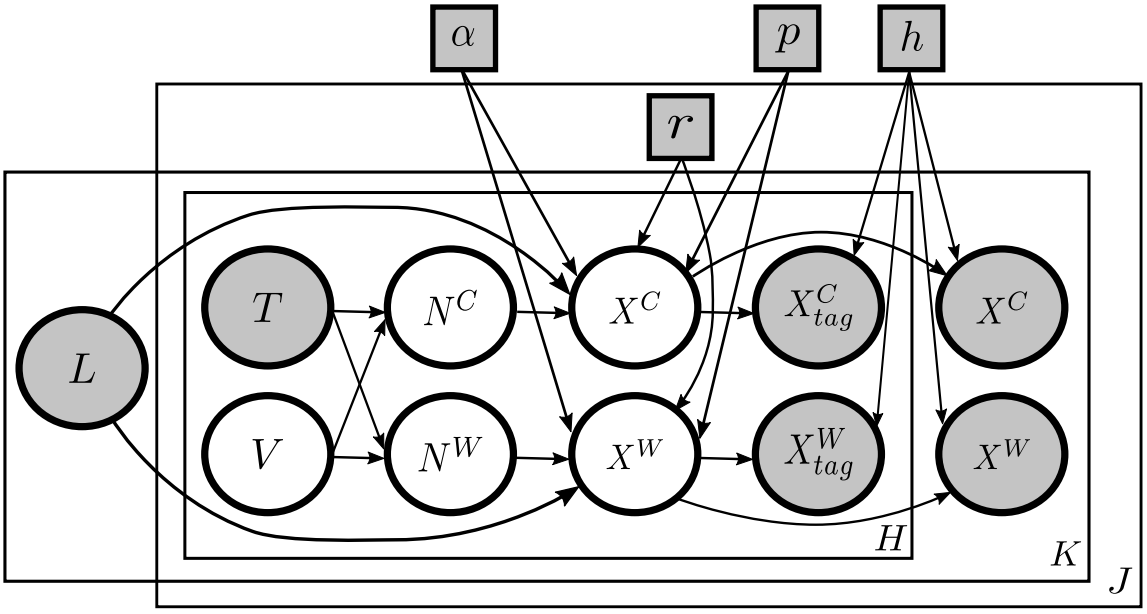


Figure 3.4: Probabilistic graphical model for SV classification. Graphical representation of the Bayesian model used for haplotype-aware SV discovery in single cells. This graphical model adopts the common plate notation: Circles represent random variables, squares show the model parameters, gray (white) objects show observed (latent) variables, arrows indicate dependencies, and large rectangles indicate that the enclosed variables exist multiple times. The model describes single cells, segments, and haplotypes. Random variables: segment length L, ground state T, haplotype-resolved SV status V (to be inferred), copy-numbers of C/W reads N^C/N^W, read counts in C/W direction X^C/X^W, and read counts in C/W direction tagged by haplotype X^C_{tag}/X^W_{tag}. Note that not all reads (counts) are observed by their haplotypes (white circles inside the H box), while they are observed with no haplotype information (gray circles outside the H box). The fraction of reads that overlap with a heterozygous SNV are observed by haplotype (tagged gray read count variables inside the H box). Model parameters: the fraction of background reads α, NB parameters p, r, and the heterozygosity rate h.

which are sparse chromosome-wide haplotype alleles.

3.3.7 Bayesian model for SV classification

Every haplotype-resolved SV implies a particular segment strand pattern for each single-cell strand state WC, CW, WW, and CC (as illustrated in Section 3.2). Namely, phased strand states and structural variants define a unique W and C copy number for a genomic interval in each single cell.

Note that segment strand pattern is different from single-cell strand states. The latter refers to the inherited strand state in a single cell in a chromosome, which can be variable in the range of a chromosome in case an SCE event happens, and it can only have one of the four possible states in $T = \{CC, CW, WC, WW\}$ for diploid cells. However, segment strand pattern is a function of single-cell strand state and the

SV class in a segment. For example, if the strand state in a single cell is WW in a chromosome and an inverted duplication happens in a segment in that chromosome, the strand pattern in that segment will be WWC. Additionally, examples in Figure 3.1 show different SV examples in a single cell with WC strand state. The resulting strand patterns with different SVs at haplotype 2 are W for deletion, WCC for duplication, WW for inversion, and WWC for inverted duplication.

Based on the segment strand patterns resulting from a structural variant, we can compute posterior probabilities of single-cell structural variants resolved by their haplotypes. We developed a Bayesian statistical framework for computing these posterior probabilities. The Bayesian framework consists of two probabilistic models: a negative binomial (NB) model for the total single-cell strand-specific read counts and a multinomial model for haplotagged W/C read counts in single cells (see Figure 3.2.d).

Figure 3.4 shows a graphical representation of the Bayesian model, called plate notation. As illustrated in the figure, each segment and haplotype of a single-cell genome comes with variables V for the SV state and T for the ground strand state, which we refer to as $V_{j,k,h}$ and $T_{j,k,h}$, respectively. Note that j, k, h indices are not written as part of variable names in the plate notation, and they are, according to conventions of plate notation, equivalent to the repetition numbers in the bottom right corners of the plates. However, in the text notation, we use j, k, h for random variable indices as indicators for single cells, segments, and haplotypes, respectively. A haplotype-resolved SV and strand state deterministically lead to a corresponding copy number observed in Crick and Watson direction, as explained in Section 3.2.

The Bayesian model parameters include the fraction of background reads α, the NB parameters p and r, and the heterozygosity rate h. More precisely, α represents noise in the direction of Strand-seq reads that can happen for example in regions with incomplete BrdU incorporation or removal [26]. Such wrongly oriented reads are denoted as *background reads* with existence probability of α. Moreover, p and r are the parameters for NB distribution, which we used for modeling Strand-seq read counts. The h parameter refers to the probability of having a heterozygous position in the genome. In the following sections, we elaborate on the details of the Bayesian model and its parameters. We will show how to estimate model parameters and compute posterior probabilities for each of the negative binomial and multinomial models.

Negative Binomial distribution for read counts

We modeled W/C strand-seq read counts by a negative binomial (NB) distribution. The NB distribution captures the overdispersion problem resulting from non-uniform coverage of sequencing data and has been used to model genomic read counts [64]. Let us denote the value n as the number of single cells analyzed in a sample. We assume that the number of reads sampled from each single cell at a fixed bin size is an NB random variable. In reality, the coverage of single cells will be varying resulting in different NB parameters for each cell. We assume that all single cells have the same p parameter, therefore there are n+1 free parameters to estimate (one p parameter and n dispersion parameters).

Having the same p parameter over all single cells implies that the ratio of mean to variance is constant across all single cells. This assumption implies that that in the

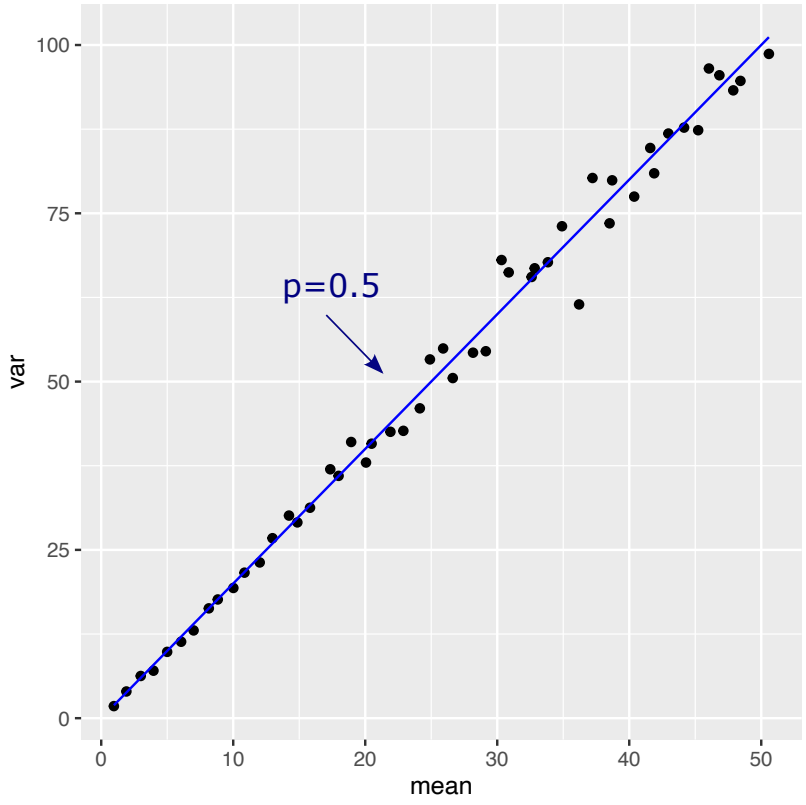


Figure 3.5: Example of linear mean-variance relationship in the NB distribution. Scatter plot of mean-variance relationship between random NB samples with $p = 0.5$ and different r parameters from 10 to 50. Every point in the scatter plot shows the empirical mean (x-axis) and variance (y-axis) in a set of 1000 *iid* random integers sampled from an NB distribution with $p = 0.5$. Different points correspond to mean-variance of NB distributions with various dispersion parameters $1 \leq r \leq 50$.

NB distribution, the ratio of the mean to the variance is equal to $1 - p$, according to equation 2.2. Consequently, mean-variance points in a set of our NB distributed random variables fit a line that passes the origin with a slope determining the p parameter. Figure 3.5 is an example of the mean-variance linear relationship in different NB distributions with $p = 0.5$. Different points in the scatter plot show the empirical mean and variance in samples of size 1000 from different NB distributions with parameters $p = 0.5$ and $1 \leq r \leq 50$. The figure shows the fitness of mean-variance points to a line that passes the origin coordinate, the slope of which determines the shared p parameter.

The aforementioned assumption in our NB distribution model implies a linear relationship between the mean and variance of binned read counts among single cells. This relationship allows estimation of the shared p parameter: for each single cell, we compute the empirical mean and variance of the observed read counts in fixed-sized bins across the genome. If we denote the set of empirical mean-variance pairs for single

cells by $(m_1, s_1^2), (m_2, s_2^2), \cdots, (m_n, s_n^2)$, the p parameter is estimated as follows:

$$p = 1 - \frac{\sum_{i=1}^{n} m_i}{\sum_{i=1}^{n} s_i^2} \tag{3.1}$$

After obtaining p, we estimate the dispersion parameter of each single cell by setting the distribution mean to the average read count per bin of that single cell. Hence the dispersion parameter of the ith cell denoted as r_i is computed as follows:

$$r_i = \frac{m_i(1-p)}{p} \tag{3.2}$$

We employed a trimmed mean for estimating the dispersion parameters (with TRIM parameter set to 0.05), to remove the effect of abnormally high or zero read counts (e.g., seen in regions of low mappability). More exactly, we exclude the set of bins with the lowest and highest 5% coverage values. We perform this task to have a more robust parameter estimation due to the existence of extremely low or high bin coverage resulting from bias factors such as low mappability.

Computing SV likelihoods in single cells

Let us define the sum of Crick and Watson copy numbers of both haplotypes as $N_{j,k}^C = N_{j,k,h_1}^C + N_{j,k,h_2}^C$ and $N_{j,k}^W = N_{j,k,h_1}^W + N_{j,k,h_2}^W$. Conditional on these Crick and Watson copy numbers, the corresponding read counts are assumed to follow an NB distribution:

$$\begin{aligned} X_{j,k}^C | N_{j,k}^C &\sim NB(r_{j,k}^C, p) \\ X_{j,k}^W | N_{j,k}^W &\sim NB(r_{j,k}^W, p) \end{aligned} \tag{3.3}$$

for each single cell j and segment k. Here, p is the estimated common p-parameter of the NB distribution, and $r_{j,k}^C$, $r_{j,k}^W$ are proportional to the estimated parameter r_j (see Section 3.3.7), the segment size L_k (in terms of the number of bins) and the Watson and Crick segment copy-numbers $N_{j,k}^C, N_{j,k}^W$. Hence these r parameters are computed as follows (for $d \in \{C, W\}$):

$$r_{j,k}^d = \begin{cases} \frac{1}{2}\alpha r_j L_k & \text{if } N_{j,k}^d = 0 \\ \frac{1}{2}(1-\alpha) r_j L_k N_{j,k}^d & \text{otherwise} \end{cases} \tag{3.4}$$

, where α is a parameter in our model indicating the fraction of background reads sequenced in the wrong direction, which represents noise in Strand-seq data (for example due to regions with incomplete BrdU incorporation or removal [26]). These background reads are taken into account by assuming $\alpha = 0.1$ reflecting an upper bound for their abundance observed in practice. Note that the $\frac{1}{2}$ coefficients in the above formula serve to scale the dispersion parameter to copy-number 1 (is estimated above to reflect a diploid state of copy-number 2). In summary, every haplotype-resolved SV class (V) in a segment together with the ground strand state (T), define a Watson and Crick copy-number (N) used to compute the NB likelihood of observed read counts.

Multinomial distribution for haplotagged read counts

One of the key advantages of scTRIP is the ability to utilize haplotype information made available through strand-specific sequencing. This haplotype-awareness is brought forth by distinguishing WC from CW ground states. Our framework is additionally able to make use of reads not directly assigned to a haplotype (i.e. those in WW and CC regions) owing to their overlap with a haplotype-phased SNV.

In the haplotype phasing step, every Strand-seq read that covers a haplotagged SNV locus is tagged with either H1 or H2, depending on which haplotype allele it covers. The resulting happlotagged read counts are incorporated in the Bayesian model as random variables X^C_{tag} and X^W_{tag} (see Figure 3.4). We employed a multinomial distribution to model the conditional distribution of these tagged read counts given the (haplotype- and strand-specific) copy-numbers N^C and N^W. More precisely, we defined parameters of the multinomial distributions p^C_{j,k,h_1}, p^C_{j,k,h_2}, p^W_{j,k,h_1}, and p^W_{j,k,h_2} for each segment k and single cell j, such that they are proportional to the corresponding copy-numbers:

$$p^d_{j,k,h} \propto \max(\alpha, N^d_{j,k,h}) \tag{3.5}$$

where $d \in \{C, W\}$ as before. Here, α is again a rate of background reads (set to 0.1) and the $p^d_{j,k,h}$ parameters are normalized to sum up to one. Given the total number of reads and the (haplotype- and strand-specific) copy-numbers N^C and N^W, the number of tagged Crick and Watson reads are multinomially distributed:

$$(X^C_{j,k,h_1,tag}, X^C_{j,k,h_2,tag}, X^W_{j,k,h_1,tag}, X^W_{j,k,h_2,tag}) \sim Multinomial(p^C_{j,k,h_1}, p^C_{j,k,h_2}, p^W_{j,k,h_1}, p^W_{j,k,h_2}) \tag{3.6}$$

Regularization and prior probabilities

In order to avoid extremely small values, we add a small constant number as regularization factor (default=10^{-6}) to all likelihoods for different SV classes in single cells and segments. In addition, we use two types of prior probabilities: The first prior is a set of constant coefficients per SV type based on the biological knowledge of observing each type. These coefficients are set to 200 for the reference (no SV) state, 100 for deletion, inversion, duplication, 90 for inverted duplication, and 1 for any other type of more complex SV. The second prior is set for each segment to cell population SV weights in order to encourage SVs present in a larger cell population. For computing these SV weights, we sum up likelihoods per SV type across all cells and normalize them to one.

Let us define $l^v_{j,k,h}$ and $P^v_{j,k,h}$ as the likelihood of structural variant $v \in \mathcal{V}$ at single cell $j \in \{1, \cdots, J\}$, segment $k \in 1, \cdots, K$, and haplotype $h \in \{H_1, H_2\}$, respectively. We also denote γ, P^v_r and P^v_{pop} as regularization factor, prior probability and cell population prior probability of structural variant v, respectively. The prior and population prior for each SV, according to the aforementioned logic, are proportional to the following

terms:

$$P_r^v \propto \begin{cases} 200 & \text{if } v = \text{ref} \\ 100 & \text{if } v \in \{\text{del, inv, dup}\} \\ 90 & \text{if } v = \text{inv-dup} \\ 1 & \text{if } v \in \mathbb{V}\backslash\{\text{ref, del, inv, dup, inv-dup}\} \end{cases}$$

$$P_{pop}^v \propto \sum_{j=1}^{J} l_{j,k,h}^v \tag{3.7}$$

The regularization is based on the assumption that with probability of γ, SVs come from uniform distribution at a segment and single cell; otherwise, it comes from the aforementioned Bayesian model, therefore the final posterior probabilities are computed using the following formula:

$$P_{j,k,h}^v \propto \frac{\gamma}{|\mathcal{V}|} + (1-\gamma) P_r^v P_{pop}^v l_{j,k,h}^v \tag{3.8}$$

Note that the exact values for each the aforementioned prior and posterior probabilities are derived by normalizing them to 1. The output of the Bayesian framework is a set of posterior probabilities for all SV types, single cells, and segments.

3.3.8 SV calling parameter settings

Our framework has 'strict' and 'lenient' SV calling parametrizations to take into account the trade-off between sensitively calling subclonal SVs present at low CF, and accurately calling SVs that are consistent among cells. The strict (default) parameter setting increases precision for SVs that happen in CF$\geq$ 5%, and the 'lenient' caller calls all SVs even those present in only one single cell. These parameterizations differ in three settings:

- The parameter *GTCUTOFF* is set to 0.05 for the strict setting, while it is set to 0 for the lenient setting. GTCUTOFF is defined as the likehood of presence of an SV genotype in the cell population. Strict parameterization requires an SV to be present in at least 5% of single cells, however lenient setting is for calling SVs present in any number of single cells.

- The haplotagged read counts information is disabled in the strict setting, while it is used in the lenient setting with the rationale that haplotagging is mostly valuable to resolve putative SVs with low CF.

- We used the filtering routine described in the previous section for the strict caller, while we proceed with the unfiltered set for the lenient caller.

Unless stated otherwise, SV calls presented in this study were generated using the 'strict' parameterization, to achieve a callset with low number of false positive SVs .

3.3.9 Post processing of SV calls

We developed a filtering routine to be used only in the strict parameterization, the main goal of which is to arrive at a high confidence SV callset for all SVs with *Cell Frequency (CF)* greater than 5%. This filtering routine removes rare inversions seen in only one or two cells, since rare inversions may occasionally correspond to SCEs. The filtering procedure further removes SV calls exhibiting particular biases, most importantly those biased to occur largely in the context of a certain ground state. In particular, while SVs can be detected in the context of all four ground states (WW, CC, WC and CW), we noticed during the development of scTRIP that artifact SV calls can occasionally arise on WW or CC chromosomes, where the ability of the caller to measure gains or losses in read depth is reduced. The following hard filters were implemented to be used with the strict parameterization:

- Removal of inversions seen in less than three cells.
- Removal of deletions seen in multiple cells, if they show a bias towards occurring mostly in WW and CC chromosomes with less than a third seen in WC or CW regions. As reasoned further above, we implemented this filter since deletions that are repeatedly seen in the WW or CC ground state, but not or only rarely in the WC ground state, are (according to our experience) of lower confidence.
- Removal of duplications seen in multiple cells, if they show a bias towards occurring in WW and CC chromosomes, with less than a third seen in WC or CW chromosomes. As reasoned further above, we implemented this filter since according to our experience duplications that are repeatedly seen in the WW or CC ground state, but not or only rarely in the WC ground state, are of lower confidence.
- Removal of SVs overlapping UCSC annotated segmental duplications in the genome by more than 50% (we found such SV calls to be of lower confidence).

3.4 Results

For investigating structural variants, single-cell Strand-seq libraries were generated for retinal pigment epithelial (RPE) cell lines including RPE-1 (RPE wild type), known for genome instability [47, 65, 66, 123], and RPE-C7 [90], which is known for cell transformation. Both of these cell lines are originated from the same female donor. The number of generated single-cell strand-seq libraries in our study were 80 and 154 for RPE-1 and C7 cell lines, respectively, with median coverage depth of 0.017× per single cell.

Among the class of deletions, duplications, inversions, and inverted duplications, scTRIP detected 54 and 53 SVs in RPE-1 and C7[1] cell lines, respectively. Among this set of SVs, 25 of them were unique to RPE-1, and 24 SVs were solely present in C7, which is a sign of sample-specific somatic SV formation. A 1.4 Mb duplication in

[1] I use the names RPE-C7 and C7 interchangeably in this thesis.

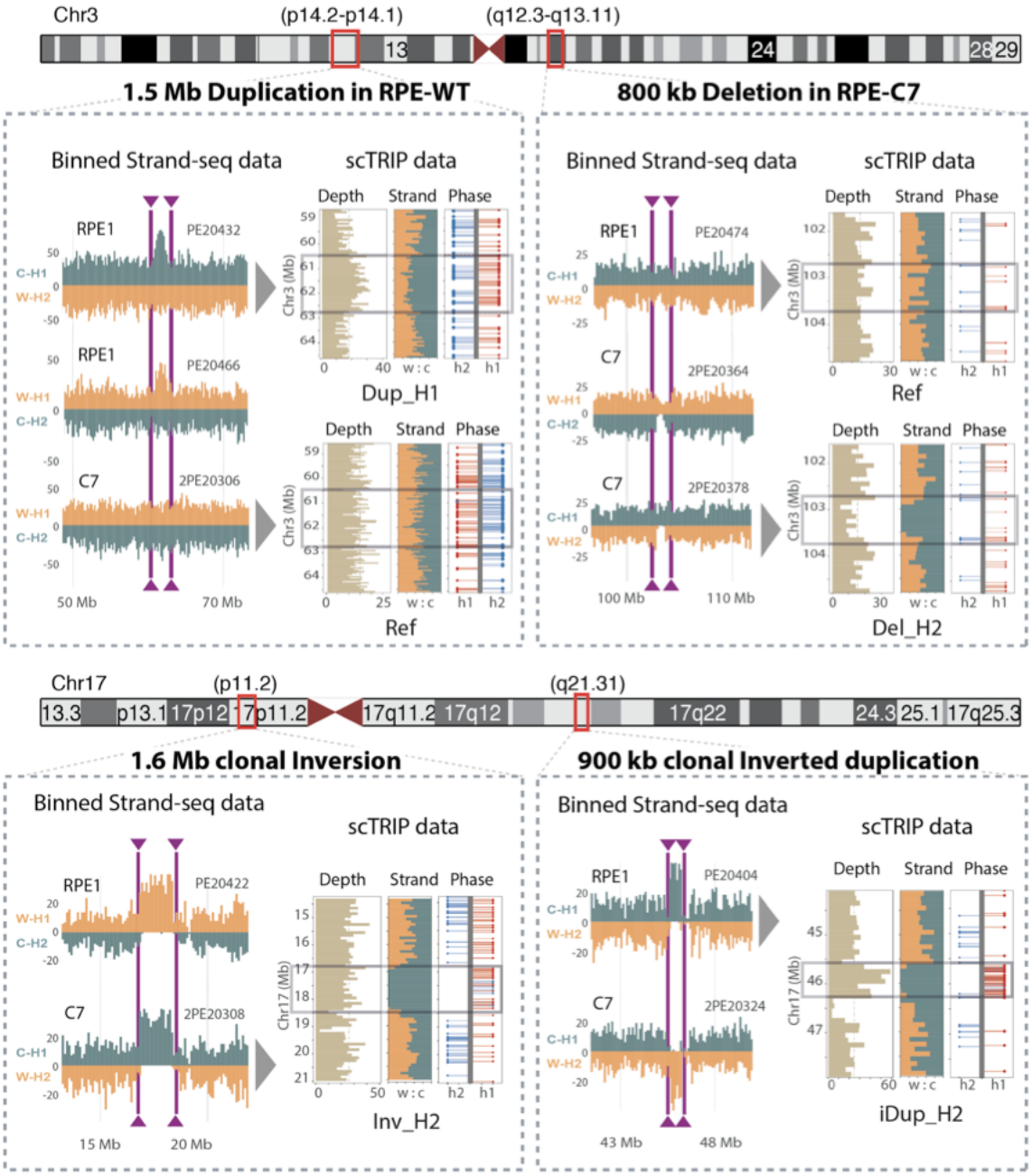


Figure 3.6: SV examples in RPE-1 and RPE-C7 cell lines. Selected examples of SV calls generated by scTRIP analysis of RPE-1 and C7 cells. For each panel, binned Strand-seq read counts separated by DNA strand (W, Watson strand [orange]; C, Crick [blue]) and haplotype (haplotype 1 [H1], above the ideogram; and H2 below the ideogram) are shown on the right, with the scTRIP layers (separated into 'Depth', 'Strand' and 'Phase') shown on the left. The box 'Depth' depicts total read counts, 'Strand' depicts the W:C fraction, and 'Phase' shows the location of haplotype-phased SNVs, with lollipop orientation reflecting the strand state of the read containing the SNV (W on left and C on right of the ideogram). Top left panel: haplotype-resolved duplication (Dup) on 3p, which is present in RPE-1 but absent in C7. Top right panel: haplotype- resolved deletion (Del) on 3q present in C7 and absent in RPE-1. Lower left panel: chromosome 17p haplotype- resolved heterozygous inversion (Inv) shared across both C7 and RPE-1. Lower right panel: chromosome 17q haplotype-resolved inverted duplications (iDup) shared in both C7 and RPE-1.

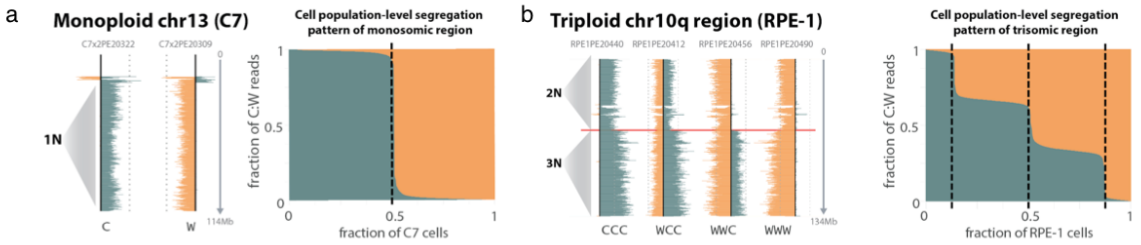


Figure 3.7: Diagnostic footprints of altered ploidy levels. For each example ploidy level, the observed strand patterns inherited by single cells are shown on the left, with the population-level segregation patterns for the respective chromosome (or locus) on the right. a) Template strand state patterns depicted are from C7, which has a monosomy chromosome (chr 13) [48, 91]. The left panel shows chr13 strand-patterns from two representative cells, with a visible 1:0 pattern characteristic for monosomic (1N) region. The right panel summarizes the fraction of observed W and C reads across n=154 sequenced cells. b) Patters of RPE-1 cells exhibiting a 10q trisomic region [48]; left panel shows strand-patterns from four single cells; right panel summarizes the fraction of W and C reads for the 10q region across n=80 cells, revealing 2:1 and 3:0 strand ratios characteristic for trisomy.

RPE-1 and a 800 kb deletion in C7 at chromosome 3 are two examples of somatic SVs (Figure 3.6a). Most of deletion and duplications were somatic and observed uniquely in RPE-1 or C7, however inversions and inverted duplications were germline SVs and mapped to known inversion polymorphisms [94]. Figure 3.6b shows an example of a 1.6 Mb germline inversion on chromosome 17. In addition to the aforementioned events, the ploidy detection tool in our framework has also successfully spotted known ploidy changes in chromosome arms, including haploid state in $13q^2$ in C7 (Figure 3.6c) and triploidy in a 10q region in RPE-1 (Figure 3.6d).

We also identified previously reported somatic chromosome arm-level CNAs, including deletion of 13q in C7 and duplication of a 10q region in RPE-1. These nondisomic regions enabled us to test our ploidy state footprints (Figure 3.7). As predicted, the 13q-arm showed a 1:0 strand ratio, diagnostic for monosomy, and the 10q region exhibited 2:1 and 3:0 strand ratios, diagnostic for trisomy.

For evaluation, we validated our set of discovered SVs present in at least 30% of cells ($CF \geq 30\%$) with bulk whole genome (WGS) data. Out of 9 SVs detected by scTRIP, all of them (100%) in C7 and 8 of them in RPE-1 (89%) were validated using WGS data. Furthermore, we ran the SV caller Delly [89] and performed read-depth analysis on bulk WGS data for both cell lines to obtain a set of SVs with at least 200kb length. By comparing this test SV set with scTRIP results, we observed that scTRIP recalled 82% of this set of SVs.

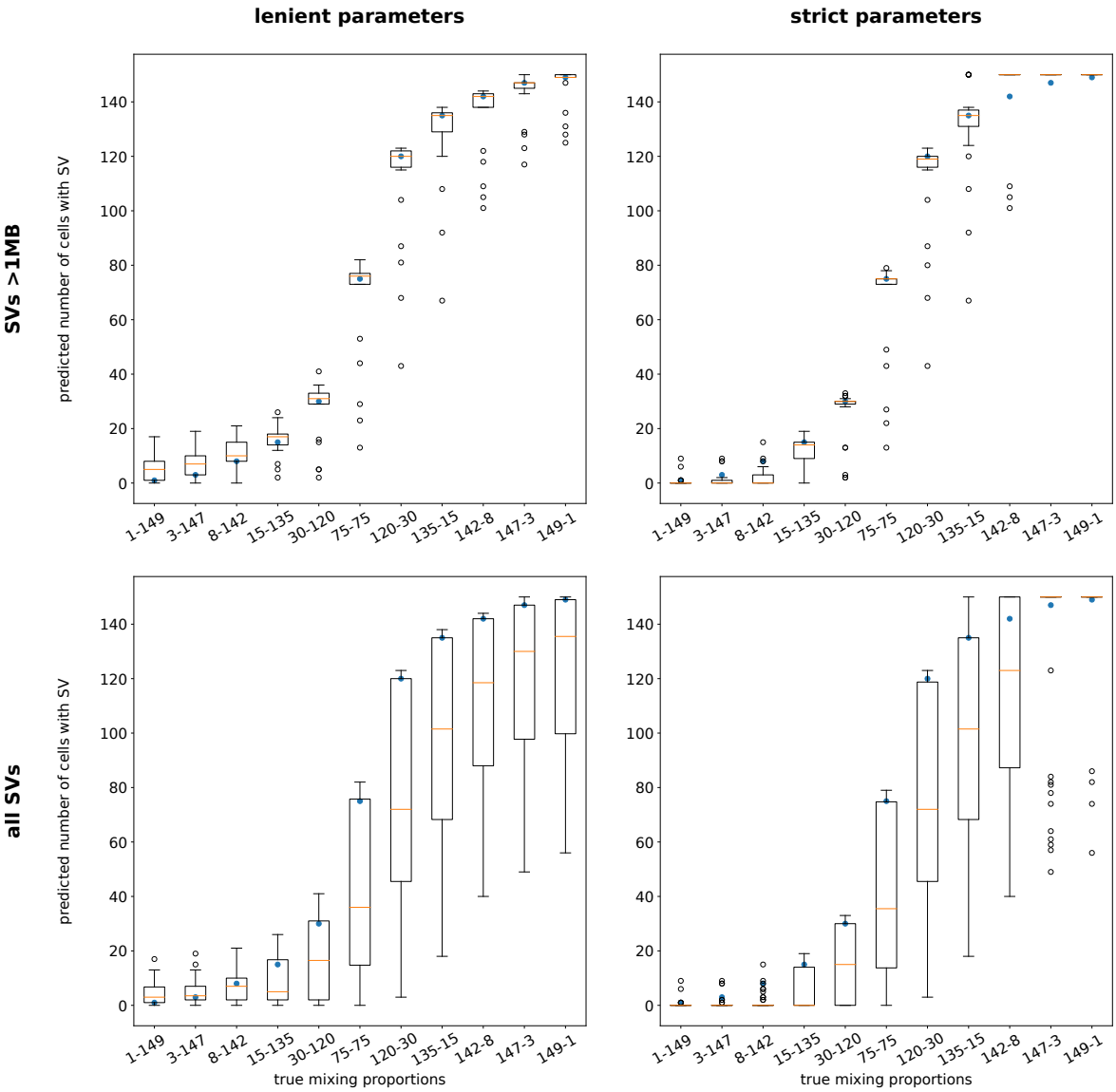


Figure 3.8: Evaluation by in silico cell mixing. SV calling performance when mixing single-cell libraries from RPE-1 and C7 in silico. Each experiment was pursued by randomly sampling n=150 single-cell libraries with replacement. Eleven different proportions of cells from RPE-1 and C7, ranging from 1-149 to 149-1 (as indicated in the x-axis), were tested for each proportion, five repetitions were run. We evaluated those SVs that had been independently validated by WGS or mate pair sequencing and called with $CF \geq 0.9$ in the original scTRIP call sets. For these SVs, boxplots depict the number of cells in which an event was recalled (y-axis) for each true mixing proportion (x-axis). A perfect SV caller would yield results corresponding to the blue circles. Top row: evaluation of SVs >1 Mb in size; bottom row: SVs of all sizes; left column: lenient parameterization; right column: strict parameterization. Box-and-whisker plot elements: center line, median; bottom of box, 25% quartile (Q1); top of box, 75% quartile (Q3); whiskers, Q1 - 1.5 IQR and Q3 + 1.5 IQR where IQR is the interquartile range = Q3 – Q1; outliers above/below the whiskers are shown as circles.

3.4.1 Cell-mixing experiments

We performed evaluations based on a simulated mixture of cell lines on real Strand-seq data. Here we aimed to assess scTRIP's ability to accurately estimate subclonal variants with particularly low cell fractions (CFs). Since confidently locating SVs with low CF is difficult to achieve using bulk sequencing (and we hence cannot build a 'ground truth' set from bulk data), we devised the following experiment: We randomly sampled 150 cells from the set of RPE-1 and RPE-C7 cells (with replacement) to establish 11 different mixing frequencies (1-149, 3-147, 8-142, 15-135, 30-120, 75-75, 120-30, 135-15, 142-8, 147-3, 149-1, where the first number gives the cell count from RPE-1 and the second number gives the cell count from C7). For each of these 11 mixing proportions, we repeated this re-sampling experiment 5 times to create a total of 55 data sets. We then ran the scTRIP pipeline on each of the 55 mixing data sets to obtain a strict and a lenient call set for each of them. For evaluation purposes, we used the pure (i.e. not mixed) call sets and determined a set of "reliable" variants present in RPE-1 but not in C7 or vice versa. Here, we defined "reliable" as calls that are (near-)clonally called with a CF $\geq 90\%$ and that are additionally validated by the respective bulk sequencing data set. For this set of reliable events, we contrasted the CF level as determined by scTRIP on the mixed data set with the ground truth CF level (corresponding to the mixing proportion by design of this experiment). In case an event was not called (e.g. because of low CF), we considered the predicted cell fraction to be 0.

Figure 3.8 shows the results of these simulation experiments. These plots reveal that for large SVs (>1 Mb), the CF estimates (shown as box plots) are generally in very good agreement with the true CF (shown as blue dots), with the limitation that the strict parameter set tends to miss calls at very low CFs, e.g. for mixing proportions 1-149 and 3-147. When including smaller events below 1 Mb in this evaluation, then the CF estimates show a larger variance (which is expected) and have a tendency towards underestimating the true CF (bottom row). This reflects the intrinsic difficulty to reliably detect short events in each individual single cell from low coverage data sets.

3.4.2 Breakage-fusion-bridge cycles

Cancer genomes frequently harbor complex DNA rearrangements that can facilitate accelerated tumor evolution [122]. One example are breakage–fusion–bridge cycles (BFBs) [9, 60, 92, 98], a chromosome instability mechanism discovered by Barbara McClintock [68, 69].

BFBs happen when replicated sister chromatids lose their telomeric parts (see Figure 3.9). Since telomeres prevent sister chromatids to fuse together, their absence in sister chromatids cause them to fuse with each other. In anaphase, a part of cell division, the fused sister chromatids form a bridge, and their centromeres are pulled into two opposite directions of the cell for separation of the replicated chromosomes and forming a new cell. Being pulled towards opposite directions causes them to break apart from each other. This breakage does not necessarily happen at the fusion point, which results in uneven sister chromatids where one of them gains an inverted copy

[2] q arm of chromosome 13

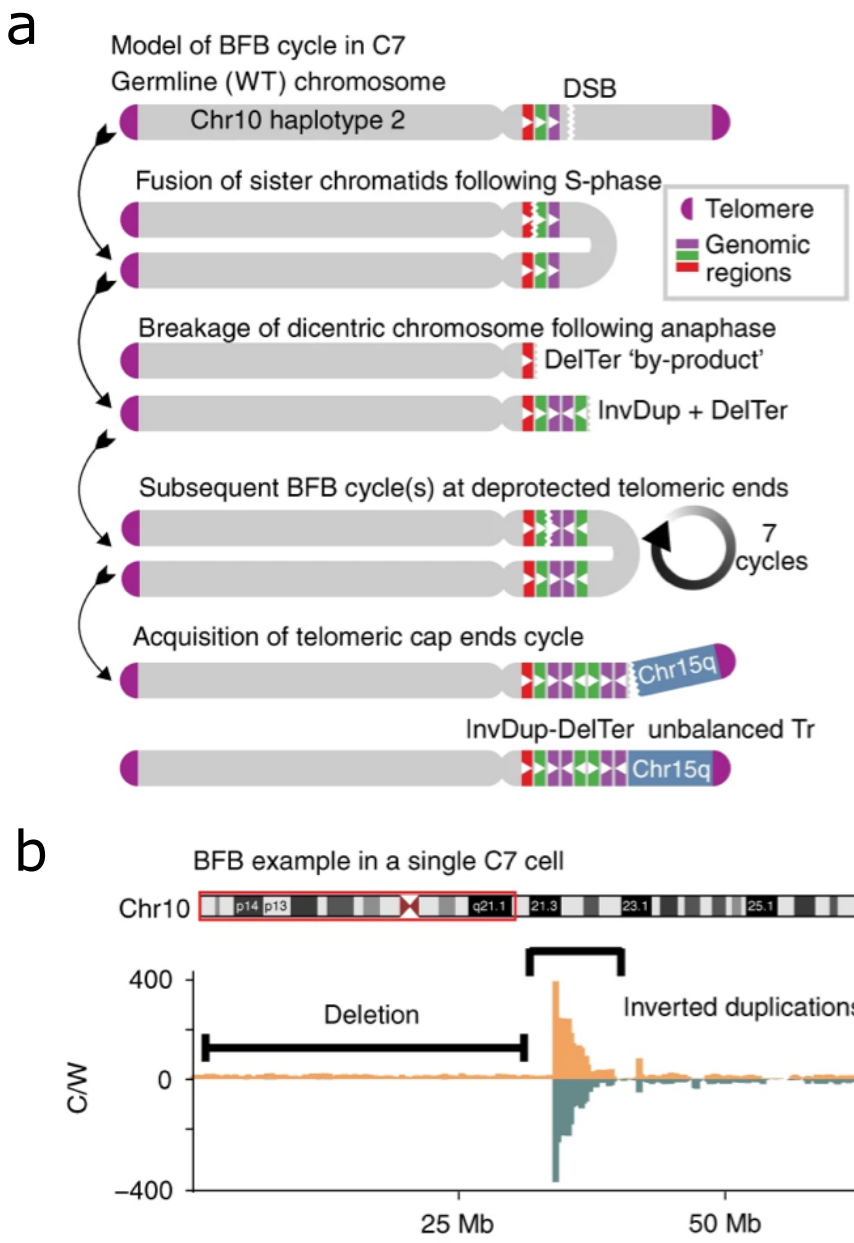


Figure 3.9: Breakage Fusion Bridge Cycle (BFB) mechanism. a, Strand-specific read depth of an example C7 cell showing a region of inverted duplication (InvDup)-mediated amplification on 10p, with adjacent terminal deletion (DelTer) of the same haplotype, resulting from BFB cycles. b, Model of the mutational process leading to the observed structures seen for the 'major clone'. Amplification via BFB cycles typically proceeds in 2^n CN steps, suggesting that approximately seven successive BFB cycles occurred. According to our model, translocation of 15q terminal sequence stabilized 10p BFB. DSB, double-strand break. WT, wild type.

of a DNA segment. This process continues in several cycles until it is stabilized by receiving a telomere from a chromosome through a translocation process.

The repetition of BFB cycles duplicate regions in inverted orientation (that is,

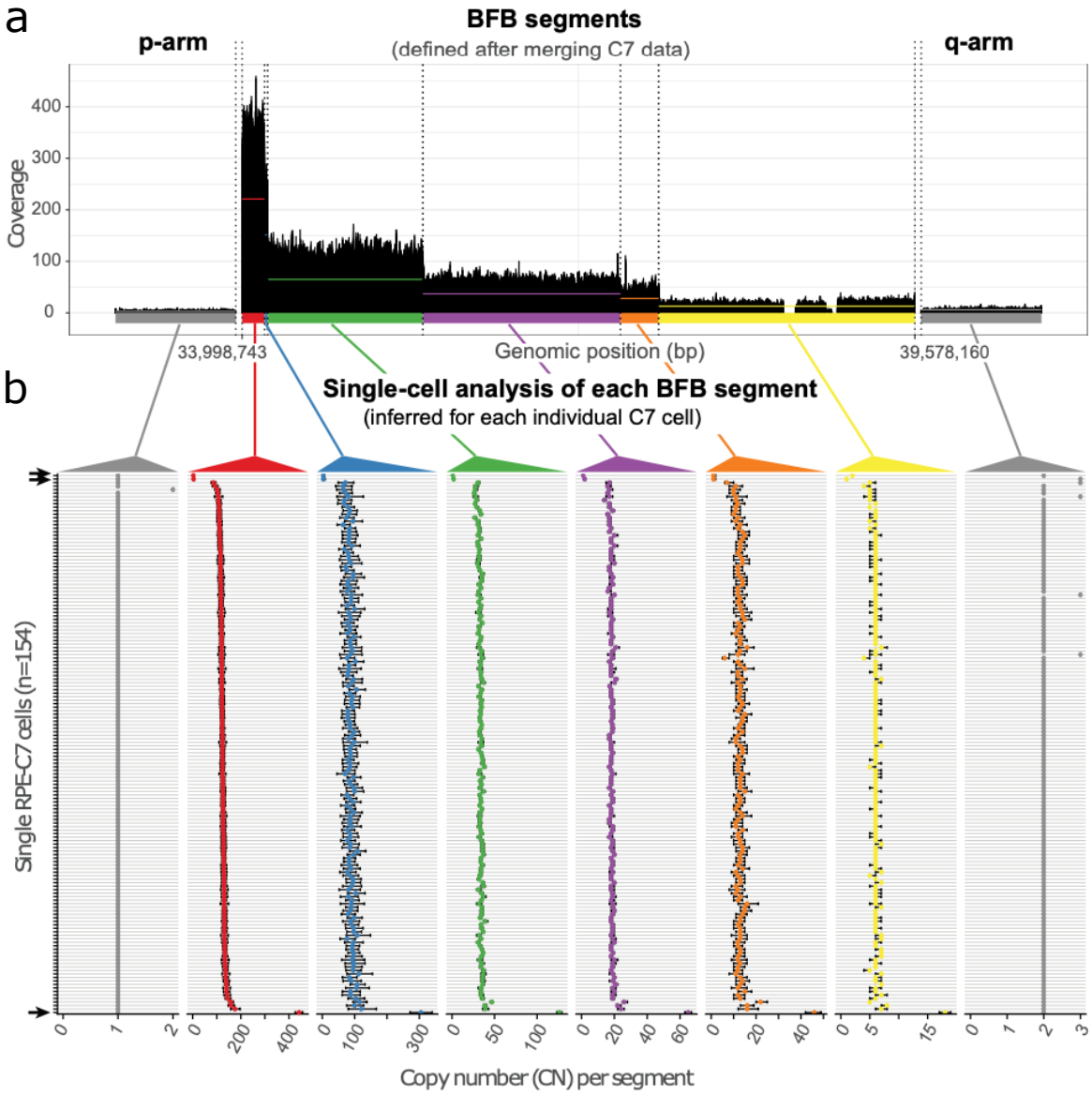


Figure 3.10: Clustering single cells in BFB cycles. a, Aggregated read data from 154 C7 cells to highlight the stepwise CN change for the 10p amplicon. Colors indicate six segments identified within the amplicon, shown by horizontal lines (mean CN per segment: red = 221, blue = 151, green = 65, purple = 37, orange = 28, yellow = 13) Gray denotes regions flanking the amplicon. b, Genetic single-cell diversity within the 10p amplicon. CN (x axis) values are shown across each individual sequenced C7 cell (y axis), to provide cell-by-cell estimates of CN for each segment defined in c.

generate InvDups) adjacent to a terminal deletion (here called 'DelTer') on the same homolog. BFBs rising to high cell fraction can be inferred from bulk WGS by locating 'fold-back inversions' from read-pair alignments [9]; however, owing to high coverage requirements this cannot be systematically achieved in single cells. We reasoned that scTRIP could provide a new opportunity to directly study BFB formation in single cells.

To investigate BFBs, we first interrogated C7, in which fold-back inversions were previously described [67]. Closer analysis of 10p showed an amplicon containing 'stepwise' InvDups with an adjacent DelTer on the same haplotype, consistent with BFBs (Figure 3.9b). scTRIP located a series of clustered InvDups on the 10p arm, detected in 152 of 154 cells (Figure 3.10a). Upon aggregating reads across cells, we identified eight discernable segments: the 10p amplicon comprising six stepwise copy number (CN) changes, the adjacent 10p terminal deletion and the centromere-proximal disomic region (Figure 3.10a). We used these eight segments to infer the cell-specific CN status for each cell (Figure 3.10b). It revealed three genetically distinct subclones: (1) 151 cells (the 'major clone') showed 'intermediate' CNs of 100–130 for the highest CN segment; (2) two cells lost the corresponding 10p region through a DelTer; and (3) one cell exhibited vastly higher CNs (440 copies) for this segment, suggesting it underwent additional BFBs.

3.5 Subclonal complex rearrangements uncovered in T-ALL

To evaluate the diagnostic value of scTRIP, we next analyzed leukemic samples. Both somatic balanced and complex SVs, which typically escape detection in single cells, are abundant in leukemia [38, 61, 103]. We analyzed a T-ALL relapse sample obtained from a juvenile female patient (P1). We sequenced 79 cells and discovered two subclones, each represented by at least 25 cells.

In our analysis of subclonal SVs in P1, we discovered a low-frequency (CF = 0.32%) series of rearrangements affecting a single 6q haplotype. Figure 3.11 shows the results of this subclonal rearrangement detection. These rearrangements comprised two Invs, one InvDup, one Dup and three Dels, resulting in 13 breakpoints spanning nearly 90 Mb (Fig. Figure 3.11b).

Our T-ALL sample analysis highlights that scTRIP might be applicable in the diagnosis of diseases with complex rearrangement and chromothripsis events, which is highly valuable in cancer evolution studies.

3.6 Data and code availability

The sequencing data for this study is available at the European Genome-phenome Archive (EGA) and the European Nucleotide Archive under the following accession numbers: PRJEB30027, PRJEB30059, PRJEB8037, PRJEB33731, EGAS00001003248, and EGAS00001003365. The patient data can be accessed by the governance of EGA Data Access Committee.

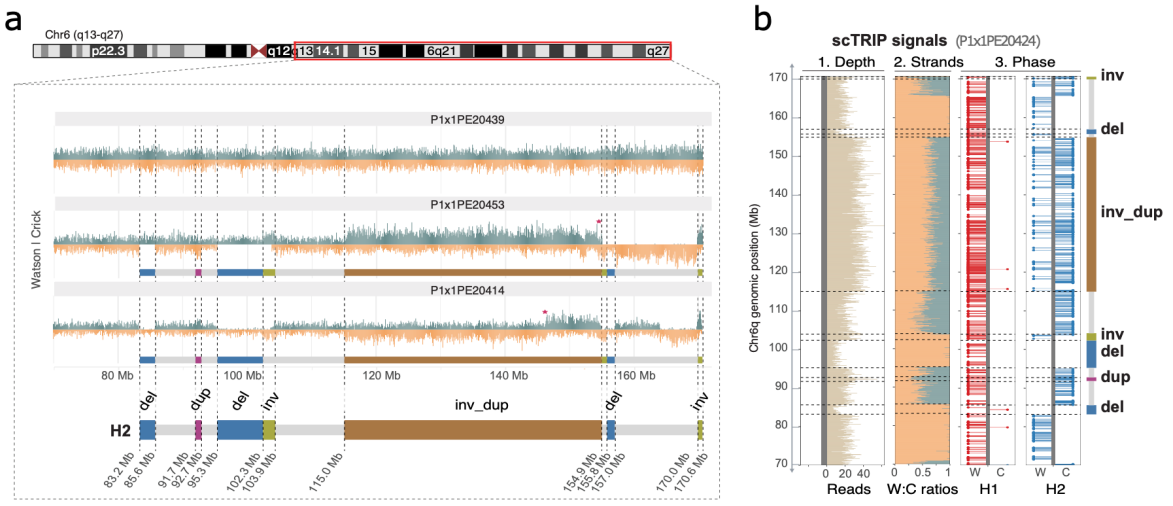


Figure 3.11: Locating SVs in a T-ALL sample. a, Reconstruction of subclonal clustered DNA rearrangements at 6q via scTRIP. Three single cells of type WC are shown as example including one normal cell and two rearranged cells. Horizontal bars show the SV calls in scTRIP including deletion, duplication, inversion, and inverted duplication. b, scTRIP's three layers signature (coverage, strand ratio, and haplotype information) leading to haplotype-resolved analysis of SVs clustered at 6q, all of which fall onto H2.

The source code of our computational framework is publicly available on GitHub at the following link addresses:

scTRIP pipeline: https://github.com/friendsofstrandseq/mosaicatcher-pipeline

Translocation detection tool: https://github.com/friendsofstrandseq/TranslocatoR

Segmentation tool: https://github.com/friendsofstrandseq/mosaicatcher

3.7 Discussion and future work

scTRIP enables systematic SV detection in single cells by integrating three complementary data layers. The combined costs are currently ∼US$15 per cell, and the protocol requires ∼2 days to generate 96 libraries. Previous single-cell studies investigating distinct SV classes involved deeply sequencing only a few cells following WGA[22, 28, 33], and previous SV detection efforts using Strand-seq were centered on germline inversions[94]. scTRIP, facilitated by our Bayesian calling framework, enables systematic discovery of a wide variety of disease-relevant somatic SV classes, including repeat-embedded SVs largely inaccessible to standard WGS in bulk. SVs detected by scTRIP are haplotype-resolved, which helps reduce false positive calls and facilitates allele-specific expression analyses[22].

We showcase how scTRIP can infer complex mutational processes by identifying sporadic BFBs in up to 8% of transformed RPE cells, revealing that somatic SV formation via BFB cycles is markedly abundant. Indeed, BFB cycles represented the

most common process for SV formation identified after chromosomal arm-level and terminal loss and gain events, all of which can result from chromosome bridges[3, 106]. BFB cycles have also been reported in cleavage-stage embryos from in vitro fertilization (revealed by hybridization-based single-cell assays)[108] and occur in a wide variety of cancers [55], can precipitate chromothripsis [60] and correlate with disease prognosis [110].

An estimated 20% of somatic deletions and more than 50% of all somatic SVs in cancer genomes arise from complex rearrangements. By directly measuring these events in single cells, scTRIP can facilitate investigating their role in cancer evolution. Our study on a sample patient with T cell acute lymphoblastic leukemia (T-ALL) showed scTRIP's ability in uncovering low-frequency rearrangements affecting a single 6q haplotype. This experiment highlights scTRIP's utility for disease prognosis involving complex chromothripsis events. Another potential application is in rare disease genetics, where scTRIP may help resolve 'unclear cases' by widening the spectrum of accessible SVs, leading to somatic mosaicism [8]. Finally, scTRIP could be used to assess genome integrity in conjunction with cell therapy, gene therapy and therapeutic CRISPR-Cas9 editing, which can result in unanticipated SVs [54, 118].

Our approach enables the study of somatic SV landscapes with much less sequence coverage than WGA-based methods. We demonstrated SV discovery using $\sim 2,000-$fold fewer reads than required for read-pair or split-read-based methods [50]. Single-cell sequencing to deep coverage using WGA can map SVs of up to 200 kb in size and remains useful for detecting small CNAs. However, WGA-based single-cell SV analyses are subject to the limitations of paired-end analyses, allelic dropouts, and low sensitivity in repetitive regions[33]. Low depth methods for CNA profiling single cells exist and can detect CNAs of 1 to 5 Mb [33, 120]. These show promise for investigating subclonal structure in non-dividing cancer cells, harboring large CNAs, but miss key SV classes and fail to discriminate between SV formation processes.

In conclusion, scTRIP enables systematic SV landscape studies to directly investigate SV formation in single cells. It provides important value over existing methods and opens avenues in single-cell analysis.

Chapter 4

SaaRclust: clustering long sequencing reads by chromosome

This chapter represents SaaRclust, a tool for clustering long sequencing reads by their original chromosome using single-cell Strand-seq data. In this project, I shared first authorship (perceived as equal contribution) with David Porubsky under supervision of Tobias Marschall. The results of this study has been accepted in ISMB 2018 conference and was published in Bioinformatics [37]. The content of this chapter is reused from our published article.

My main contribution in this project was development of the EM clustering algorithm, under supervision of Tobias Marschall. Development of the main R package for SaaRclust was performed in a pair programming manner in which David Porubsky was the driver programmer, who wrote the code, and I was the observer. The data analysis experiments, including optimizing read mapping and SaaRclust parameters, was mainly done by David Porubsky. I did the main single cell sampling experiments and runtime analysis. I wrote the first draft of paper manuscript, and the other authors revised and approved it.

4.1 Introduction

Current sequencing technologies are able to produce reads orders of magnitude longer than ever possible before. Long read sequencing technologies, such as marketed by Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT), can produce reads of tens of kilobases in length. This allows for much improved genome assemblies in comparison to short read (Illumina) sequencing platforms [14, 42, 52, 62, 75]. In particular, long reads can resolve many repetitive regions that are inaccessible to short reads, which yields more accurate and contiguous assemblies [107]. Such long reads have sparked a new interest in *de novo* genome assembly, which removes reference biases inherent to re-sequencing approaches and allows for a direct characterization of complex genomic variants. However, assembling a mammalian genome from noisy long reads incurs a significant computational burden.

The knowledge of chromosomal origin can avoid potential misassemblies and entail substantial computational advantages: if reads were sorted by chromosome, genome

assembly could then be performed separately per chromosome, which has the potential of saving large amounts of runtime and memory, as well as improving parallelization.

Strand-seq technology has been used to cluster *contigs* into their original chromosomes, as proposed in BAIT and ContiBAIT tools [43, 81]. These tools rely on mapping Strand-seq reads first to a contig-stage assembly, and then using the strand states of the contigs to scaffold them into chromosomes [44]. The major limitation of this approach is that any assembly errors, such as chimeric contigs, result in mixed states that confound the clustering method. Additionally, this method is not designed to work with the extremely sparse data resulting from mapping Strand-seq reads to individual long reads.

In this study, we show how single-cell template Strand-seq data can be leveraged for clustering long reads *in silico*, *without* using a reference genome and *before* genome assembly. We introduce a novel latent variable model and a corresponding Expectation Maximization (EM) algorithm, termed SaaRclust, and demonstrate its ability to reliably cluster long reads by chromosome. For each long read, this approach produces a posterior probability distribution over all chromosomes of origin and read directionalities. In this way, it allows to assess the amount of uncertainty inherent to sparse Strand-seq data on the level of individual reads. To our knowledge, SaaRclust is the first approach for the *in silico* separation of long reads by chromosome prior to assembly.

4.1.1 Idea

Let us assume we are given a set of long sequencing reads, e.g., PacBio reads. As shown in Figure 4.1a, we map all Strand-seq reads to all PacBio reads and then count the number of Strand-seq reads from different libraries that are mapped to each PacBio read in either Watson (-) or Crick (+) orientations. This read-to-read mapping does not involve a reference genome. As a central observation, we note that PacBio reads originating from the same chromosome will show the same strand states across the different single-cell libraries. Therefore, we can use these directional Strand-seq mapped read counts in order to cluster PacBio reads into their original chromosomes.

The main idea of the EM algorithm, as shown in Figure 4.1b, is that the knowledge of Strand-seq strand states for each chromosome are informative of the chromosomal origin of PacBio reads, and vice versa; that is, knowing the true chromosomes of PacBio reads enables us to find the chromosome strand states. This flow of information can be repeated in an iterative manner, starting from an arbitrary initialization, using an EM algorithm. We model the process of sampling Strand-seq read counts from different libraries by a mixture model. Clustering then commences through an EM algorithm that iteratively estimates strand state parameters, cluster weights, and read assignments to clusters.

4.2 Mixture model and the EM algorithm

We consider two clusters per chromosome corresponding to PacBio reads oriented in forward and backward direction, respectively. Let N, J, K be the number of PacBio

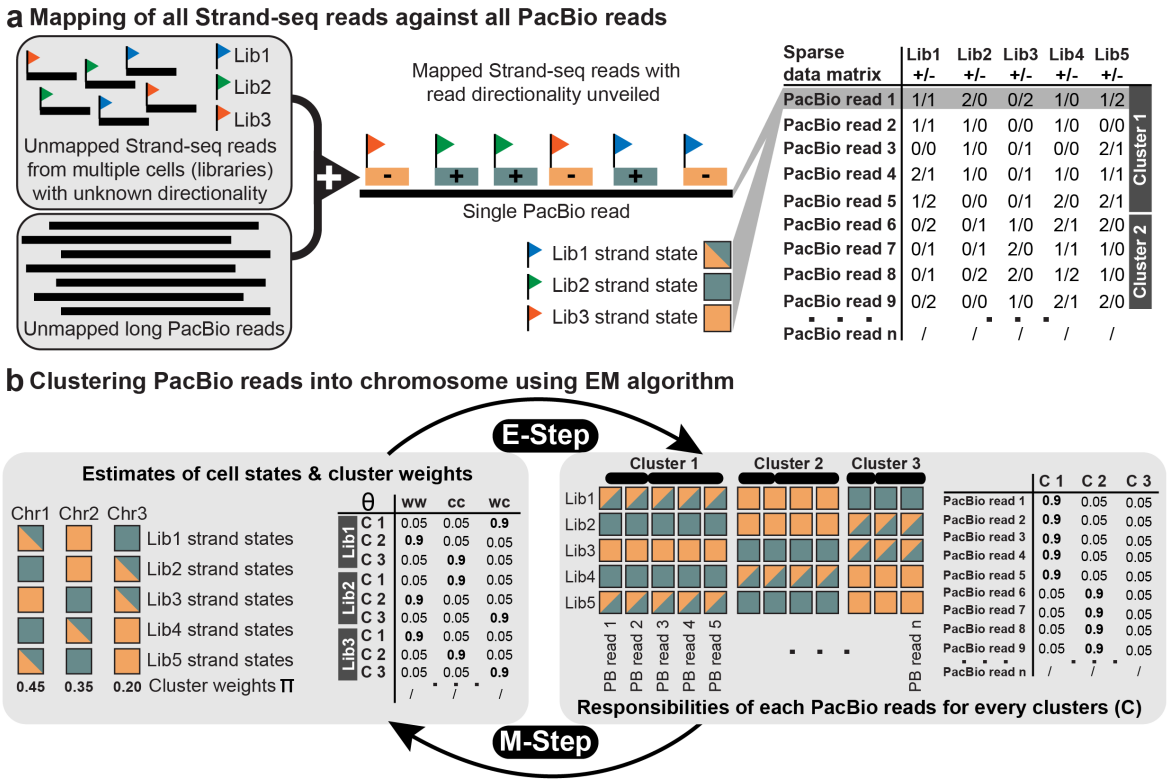


Figure 4.1: Overview on algorithmic workflow. a) After mapping Strand-seq to PacBio reads, their (relative) directionality with respect to the PacBio reads is recorded. That is, for each PacBio read and each Strand-seq single-cell library, the number of Crick (+) and Watson (-) reads is tabulated (right). For example, "2/0" refers to two Crick reads and one Watson read mapped to the corresponding PacBio read in a given row. Note that the data is sparse, with many zero entries in the table. This table is the input to our EM clustering method. b) Illustration of the main idea of the EM algorithm, which iterates between E-step and M-step. On the left, a table of chromosomal strand states probabilities (Θ) is shown, which contains the current estimates of a certain strand state (i.e. WW, CC, or WC) for each single-cell library (Lib 1, Lib 2, Lib 3) and chromosome (C 1, C 2, C 3). On the right, we illustrate that PacBio reads in the same cluster (chromosome) display the same strand signatures (in terms of the Strand-seq reads mapped to them); the table shows, for each PacBio read, the probabilities of stemming from a given chromosome (C 1, C 2, C 3). In the E-step, the current estimates of chromosomal strand states probabilities (Θ) are used to estimate cluster assignments. In the M-step, the current (probabilistic) cluster assignments are used to estimate strand state probabilities.

reads, single-cell libraries, and clusters respectively. We present a full list of notations that we use throughout this chapter in Table 4.1.

We model the number of Watson and Crick Strand-seq reads mapped to PacBio reads by a mixture model, shown in plate notation in Figure 4.2. The component weights of the mixture model are $\Pi = (\pi_1, \cdots, \pi_K)$, which are the probabilities of

Table 4.1: Overview of notations

Notation	Definition
$X^W_{n,j}$	The number of Strand-seq reads from single cell j mapped to the nth PacBio read in Watson direction
$X^C_{n,j}$	The number of Strand-seq reads from single cell j mapped to the nth PacBio read in Crick direction
$X_{n,j}$	$(X^W_{n,j}, X^C_{n,j})$
X^C_n	$(X^C_{n,1}, \cdots, X^C_{n,J})$
X^W_n	$(X^W_{n,1}, \cdots, X^W_{n,J})$
X_n	$(X_{n,1}, \cdots, X_{n,J})$
T	The set of all possible strand states $\{WW, WC, CC\}$
$t_{k,j} \in T$	The state of single cell j in cluster k
t_k	$(t_{k,1}, \cdots, t_{k,J})$
$\theta_{k,j,t}$	The probability that single cell j has state t in cluster k
$\theta_{k,j}$	$(\theta_{k,j,WW}, \theta_{k,j,WC}, \theta_{k,j,CC})$
θ_k	$(\theta_{k1}, \cdots, \theta_{kJ})$
π_k	The probability that a PacBio read comes from cluster k
$Z_{n,k}$	A binary random variable showing whether PacBio read n comes from cluster k
$[1:a]$	The set of all integers between 1 and a

sampling PacBio reads from different clusters. For every cluster $k \in [1:K]$, we have a matrix $\theta_{J\times 3}$ containing the strand state probabilities for different single cells in the specific cluster. Each row in this matrix corresponds to a single cell showing the probabilities of different strand states $T = \{WW, WC, CC\}$, so each row sums up to 1. One should note that a single cell may have more than one state in a cluster because of SCE events. In this case, the strand state changes along a chromosome, and the corresponding row of the matrix should have at least two non-negligible values. To sum up, there are two sets of parameters in the mixture model: cluster weights $\Pi = (\pi_1, \ldots, \pi_K)$ and strand state parameters Θ, which have the following constraints based on their definitions:

$$\begin{aligned} &\sum_{k=1}^{K} \pi_k = 1 \\ &\forall (k,j) \in [1:K] \times [1:J], \ \sum_{t\in T} \theta_{k,j,t} = 1, \end{aligned} \tag{4.1}$$

where $\theta_{k,j,t}$ denotes the probability that single cell j has state t in cluster k, as defined in Table 4.1.

According to Figure 4.2, for the n-th PacBio, a cluster Z_n is first chosen based on the discrete distribution Π. Then, based on the chosen cluster, a vector t_n of size J containing strand states for different cells is generated based on the strand state probabilities θ_k in the chosen cluster $k = Z_n$. At the end, given the strand states, a random matrix X_n of size $J \times 2$ containing pairs of Watson and Crick read counts for

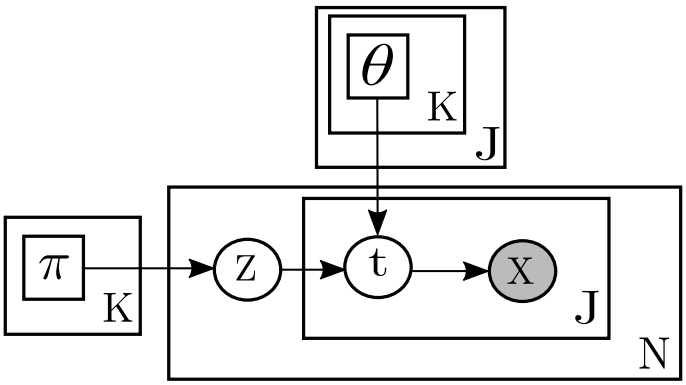


Figure 4.2: SaaRclust's mixture model expressed in plate notation. π_k denotes the weight (relative size) of cluster k. $\theta_{k,j}$ denotes a discrete probability distribution over three different strand states of single cell j in cluster k. Z_n and $t_{n,j}$ are respectively the chosen cluster and the chosen strand state of single cell j for PacBio read n. $X_{n,j}$ is a pair of Watson and Crick Strand-seq read counts of single cell j for PacBio n.

each single cell is generated by a binomial distribution. More precisely, the likelihood of observing a Watson and Crick read count, given a certain strand state t is computed as follows:

$$\mathrm{P}(X^W_{n,j}, X^C_{n,j} \mid t_{k,j} = t) = \binom{m_{n,j}}{X^W_{n,j}} p_t^{X^W_{n,j}} (1 - p_t)^{X^C_{n,j}}, \tag{4.2}$$

where $m_{n,j}$ is the total number of Strand-seq reads from library j mapped to read n (and therefore $X^W_{n,j} + X^C_{n,j} = m_{n,j}$, which we consider to be a constant) and p_t is the probability of having a Watson read from a single cell with state t is defined as follows:

$$p_t = \begin{cases} 1 - \alpha & \text{if } t = WW \\ 0.5 & \text{if } t = WC \\ \alpha & \text{if } t = CC \end{cases}$$

In the above definition, α is the fraction of background reads (reads in the opposite direction of the strand state) in WW or CC strand states, which is considered as a constant parameter in our model. In the rest of the manuscript, we abbreviate $\mathrm{P}(X^W_{n,j}, X^C_{n,j} \mid t_{k,j} = t)$ as $\mathcal{B}_t(X_{n,j})$. The likelihood of the mixture model parameters given the observed Strand-seq read counts for all PacBio reads can be then computed as follows:

$$\begin{aligned} \mathcal{L}(\theta, \pi; X) &= \prod_{n=1}^{N} \Big(\sum_{k=1}^{K} \pi_k \prod_{j=1}^{J} \big(\sum_{t \in T} \theta_{k,j,t} \mathcal{B}_t(X_{n,j}) \big) \Big) \\ \Rightarrow \log \mathcal{L}(\theta, \pi; X) &= \sum_{n=1}^{N} \log \Big(\sum_{k=1}^{K} \pi_k \prod_{j=1}^{J} \big(\sum_{t \in T} \theta_{k,j,t} \mathcal{B}_t(X_{n,j}) \big) \Big) \end{aligned} \tag{4.3}$$

The maximum likelihood problem is maximizing the objective function $\log \mathcal{L}(\theta, \pi; X)$ (log-likelihood function) in the above formula. This maximization problem does not have a closed form solution, therefore we use the EM algorithm for solving this problem, which has been shown to converge to a local optimum [116]. In order to have a simple form complete-data log-likelihood function (likelihood of the mixture model parameters given both hidden and observed random variables), we define the hidden random variables of the EM algorithm as follows: for every $(n, k, j, t) \in [1 : N] \times [1 : K] \times [1 : J] \times T$, we define a hidden binary random variable $Z_{n,k,j,t}$ which is equal to 1 if and only if PacBio read n belongs to cluster k and stems from a locus where the single cell j has strand state t. Based on this definition, there are some constraints on the hidden random variables: for every $n \in [1 : N]$, there is only one cluster $k' \in [1 : K]$ (where that PacBio read belongs to) such that the following conditions hold.

$$\begin{aligned} &\forall j \in [1 : J]; \sum_{t \in T} Z_{n,j,k',t} = 1 \\ &\forall (j, k, t) \in [1 : J] \times ([1 : K] \backslash \{k'\}) \times T; Z_{n,j,k,t} = 0 \end{aligned} \tag{4.4}$$

The complete-data log-likelihood function is computed as follows:

$$\begin{aligned} &\ln \mathcal{L}(\theta, \pi; X, Z) = \\ &\sum_{n,k,j,t} Z_{n,k,j,t} (\frac{1}{J} \ln \pi_k + \ln \theta_{k,j,t} + \ln \mathcal{B}_t(X_{n,j})) \end{aligned} \tag{4.5}$$

The EM algorithm iterates over the two following steps [20]:

$$\begin{aligned} &Q(\theta, \pi \mid \theta^{(m)}, \pi^{(m)}) = \mathrm{E}_{Z|X,\theta^{(m)},\pi^{(m)}} \ln \mathcal{L}(\theta, \pi; X, Z) \quad (E) \\ &\theta^{(m+1)}, \pi^{(m+1)} = \arg\max_{\theta,\pi} Q(\theta, \pi \mid \theta^{(m)}, \pi^{(m)}) \qquad (M) \end{aligned} \tag{4.6}$$

Let $\gamma^{(m)}(Z_{n,k,j,t})$ denote the expectation of the hidden random variable $Z_{n,k,j,t}$ given the observed data and the model parameters at the mth iteration. This expectation can be computed as follows:

$$\gamma^{(m)}(Z_{n,k,j,t}) := \frac{(\pi_k^{(m)})^{(\frac{1}{J})} \theta_{k,j,t}^{(m)} \mathcal{B}_t(X_{n,j})}{\sum_{k'=1}^{K} \sum_{t' \in T} (\pi_{k'}^{(m)})^{(\frac{1}{J})} \theta_{jk't'}^{(m)} \mathcal{B}_{t'}(X_{n,j})} \tag{4.7}$$

Based on the Equations 4.3 to 4.7, the objective function of the EM algorithm can be written as

$$\begin{aligned} &Q(\theta, \pi \mid \theta^{(m)}, \pi^{(m)}) = \\ &\sum_{n,k,j,t} \gamma^{(m)}(Z_{n,k,j,t}) (\frac{1}{J} \ln \pi_k + \ln \theta_{k,j,t} + \ln \mathcal{B}_t(X_{n,j})) \end{aligned} \tag{4.8}$$

Maximizing the objective function in Equation 4.8 by Lagrange multipliers corresponding to the constraints in Equation 4.1 leads to the following update rules for the

parameters:

$$\pi_k^{(m+1)} = \frac{\sum_{n=1}^{N}\sum_{j=1}^{J}\sum_{t\in T}\gamma^{(m)}(Z_{n,k,j,t})}{\sum_{n=1}^{N}\sum_{k'=1}^{K}\sum_{j=1}^{J}\sum_{t\in T}\gamma^{(m)}(Z_{n,k',j,t})} \tag{4.9}$$

$$\theta_{k,j,t}^{(m+1)} = \frac{\sum_{n=1}^{N}\gamma^{(m)}(Z_{n,k,j,t})}{\sum_{t'\in T}\sum_{n=1}^{N}\gamma^{(m)}(Z_{n,k,j,t})} \tag{4.10}$$

After estimating parameters of the mixture model, the cluster assignment probabilities can be computed as follows:

$$\begin{aligned}
\mathrm{P}(Z_{n,k}=1 \mid \pi_k,\theta_k) &= \pi_k \prod_{j=1}^{J} \mathrm{P}(X_{n,j} \mid \theta_{k,j}) \\
&= \pi_k \prod_{j=1}^{J} (\sum_{t\in T} \theta_{k,j,t}\mathrm{P}(X_{n,j} \mid t_{k,j}=t)) \\
&= \pi_k \prod_{j=1}^{J} (\sum_{t\in T} \theta_{k,j,t}\mathcal{B}_t(X_{n,j})),
\end{aligned} \tag{4.11}$$

where $Z_{n,k}$ denotes a binary random variable showing whether the nth PacBio read is from cluster k, as defined in Table 4.1.

4.2.1 Initializing EM parameters

For initializing the EM parameters, we use a combination of k-means and hierarchical clustering. First, we run the k-means algorithm with a number of clusters that is higher than the target number of 46 clusters for a female human genome. Note that the number of clusters in k-means is a user parameter, and we set it to a higher number in order to avoid missing small clusters. We use the J-dimensional feature vector $\left(\frac{X_{n,j}^W - X_{n,j}^C}{X_{n,j}^W + X_{n,j}^C}\right)_{j=1}^{J}$ to encode PacBio read n. Once we have run k-means on these input vectors, we compute the single-cell strand states with maximum likelihood for each cluster. Note that in this step, we use the simplifying assumption that there is no combination of strand states (resulting from SCEs) in any pair of single cell and chromosome, which makes these maximum likelihood computations straightforward. Lastly, using these single-cell strand states as a feature vector for each cluster, we merge similar clusters to obtain the desired number of clusters based on agglomerative hierarchical clustering. At the end, we use the final clusters with their maximum likelihood single-cell strand states to initialize the EM parameters. More precisely, we set the π parameters to the relative sizes of the formed clusters, and we initialize $\theta_{k,j}$, for each cluster k and single cell j, as follows:

$$\theta_{k,j,t} = \begin{cases} 0.9 & \text{if } t = \hat{t}_{k,j} \\ 0.05 & \text{otherwise} \end{cases}$$

where $\hat{t}_{k,j}$ is the estimated strand state in cluster k and single cell j.

4.2.2 Pairing clusters with the same chromosome

There are two clusters per chromosome corresponding to the PacBio reads having forward or backward direction, respectively. The directionality of mapped Strand-seq reads is exactly the opposite in a pair of clusters corresponding to PacBio reads in forward or backward direction on a chromosome. As a result, WC strand states are similar in the aforementioned pair of clusters, but WW and CC strand states are the opposite over all single cells. Based on this relation between the strand states for the clusters coming from the same chromosome, we defined a distance measure d over all pairs of clusters as follows:

$$d(clust_{k_1}, clust_{k_2}) = \sqrt{\sum_{j=1}^{J} (\theta_{k_1,j,WW} - \theta_{k_2,j,CC})^2} \tag{4.12}$$

To convert this distance measure to a similarity measure, we subtracted each computed pairwise distance from the maximum of all pairwise distances. We then used the maximum matching algorithm to find the pairs of clusters with the maximum similarities.

4.3 Experimental setup

We evaluated the performance of SaaRclust on the human female individual NA12878. The fastq files of 132 Strand-seq libraries for this individual are publicly available at the European Nucleotide Archive (ENA; `http://www.ebi.ac.uk/ena`) under accession number PRJEB14185. Additionally, raw fastq reads for all Strand-seq libraries used in this study are available at Zenodo (doi: `10.5281/zenodo.1203703`). PacBio reads are available from the Sequence Reads Archive (SRA; `https://www.ncbi.nlm.nih.gov/sra/`) under accession number SRX1837675[1]. For our study, we used the corresponding BAM file made available by Pacific Biosciences[2]. We extracted all reads from this BAM file, including unmapped ones, without applying any filters, to ensure that the reads correspond to the raw data. We reverse complemented each read mapped in reverse orientation such that all reads reflect the original direction present in raw reads. We stored the original genome mapping location as well as the mapping directionality for evaluation purposes, but did not use this information in any other way. In case of Strand-seq reads, we exported only the first mates of each read pair into fastq files. We decided not to use the second mates since the Minimap tool [56], the aligner we used in our analysis, does not handle paired-end alignment. Out of all extracted PacBio reads, those of length at least 10kb were exported as a fasta file to be used for the clustering. Since all reads in the original BAM files were sorted according to the genomic position, we have randomly shuffled Strand-seq and PacBio reads before exporting them into a fastq or fasta file, respectively.

[1] We thank Tina Graves and Rick Wilson for making this data set available.

[2] `https://downloads.pacbcloud.com/public/dataset/na12878/hg38.NA12878-WashU.bam`

4.3.1 Mapping Strand-seq reads to PacBio reads

Mapping of short Strand-seq reads to the long PacBio reads was done using the Minimap tool [56]. To allow parallel processing, we split PacBio reads into equally sized chunks of 50,000 reads per chunk. Minimap alignment was then performed on multiple chunks in parallel. We explored different parameter settings for minimap alignment, and we set the optimum parameter setting as follows: `-t 8` (number of threads), `-w 1` (minimizer window size, 1 means all k-mers are considered for high sensitivity), `-k 15` (k-mer size), `-L 50` (minimum number of matching bases per alignment), and `-f 0.05` (fraction of repetitive minimizers to be removed).

The total number of PacBio reads for individual NA12878 was 20.7M, out of which 5.8% were unmapped. After filtering them based on the minimum length of 10kb, we processed 10.8M PacBio reads, which were split in 217 chunks in total. By using Minimap, we obtained 9.1M PacBio reads with at least one Strand-seq read mapped to them.

4.3.2 Performance metrics

Original chromosomes and directionality of PacBio reads based on their mapping to the reference genome was used as a ground truth for accuracy assessment of our method. In the evaluation process, we used only the set of PacBio reads for which a ground truth was available, that is, those reads mapped to one of the autosomes and Chromosome X in the original BAM file. Note that the clustering proceeded on all reads, including unmapped ones, but the assignment of those unmapped reads cannot be evaluated.

To evaluate clustering accuracy, we first divided PacBio reads with respect to their true known chromosome and directionality. For each chromosome and orientation we assigned as true cluster the one which contained the majority of respective PacBio reads. Given this assignment, we computed the fraction of PacBio reads that were correctly assigned to their original chromosome and orientation. Such evaluation was used for hard as well as for EM soft clustering. In case of EM soft clustering, we assign each PacBio read to the cluster with highest posterior probability.

4.3.3 Hard clustering settings

For hard clustering, we selected a set of 50,000 PacBio reads that are represented in at least 35 Strand-seq libraries, i.e., the PacBio reads that have Strand-seq reads mapped to them from at least 35 different Strand-seq libraries. Such strict filtering criteria proved favorable to obtain good cluster centers using hard clustering.

To do hard clustering, we used k-means on the aforementioned subset of PacBio reads, with 54 clusters, 100 random initializations, and 10 iterations for each initialization. After k-means, we performed hierarchical clustering to merge the resulting clusters into 47 clusters, based on the estimated single-cell strand states in the clusters. Note that we observed that the PacBio reads coming from repetitive genomic regions tend to form an extra (false) cluster. This extra cluster was estimated being WC in all libraries (which is unlikely to reflect a true cell state based on the random distribution of strand states). For this reason, we set the number of clusters to 47

instead of 46, which would be an expected number of clusters for a female human used in this study.

4.3.4 Soft (EM) clustering settings

PacBio reads with an abnormally high numbers of Strand-seq reads mapped to them might adversely affect the performance of the EM algorithm in estimating the model parameters. Those reads are likely to originate from the complex repetitive regions of the genome and therefore do not have a clean Strand-seq strand state signal. We therefore removed PacBio reads that are among the top 95% quantile based on the coverage of Strand-seq reads mapped to them.

We ran our EM algorithm on each chunk of Minimap alignments independently based on the initialization of parameters resulting from hard clustering. The number of EM iterations was set to 50 in each chunk, and the α parameter was set to 0.01.

4.3.5 Runtime and Convergence of the EM algorithm

We measured running times of different steps of our pipeline. The runtime for the alignment of Strand-seq reads to PacBio reads amounted to 414.3 CPU hours, and the runtime for hard and soft clustering were 0.87 and 400.5 CPU hours, respectively.

To confirm convergence of the algorithm, we also ran the EM algorithm with 100 iterations in 10 chunks of PacBio reads, and we obtained almost the same clustering accuracy as the default number of 50 iterations in those chunks. For example, we observed that in the 100 iterations experiment, there are 78.76% of PacBio reads with the maximum cluster assignment probability of at least 0.5, among which 92.75% were correctly clustered. The two aforementioned percentages for 50 iterations are 78.72% and 92.7%, respectively, which are almost identical to the results of 100 iterations. This observation indicates that the EM algorithm has sufficiently converged after 50 iterations.

4.4 Results

Quality control. To evaluate the overall performance aligning Strand-seq reads to PacBio reads, we looked at a number of data quality measures shown in Figure 4.3a. The leftmost histogram shows how many different Strand-seq libraries are represented per PacBio read. This metric highly depends on the number of Strand-seq libraries as well as the stringency of the mapping step. We observe that the majority of PacBio reads is covered by less then 25 (out of 132) single-cell libraries. The peak on the right stems from reads in repetitive contexts which we remove in a pre-processing step (see Section 4.3.4). The middle histogram represents the number of Strand-seq reads per PacBio read per Strand-seq library. Note that we removed the zero read counts from this statistics. That is, this historgram only shows cases in which at least one read from a Strand-seq library mapped to a given PacBio read. This plot reveals that only in a minority of cases there are two or more reads from a single Strand-seq library that cover the same PacBio read. These two histograms highlight the overall sparsity

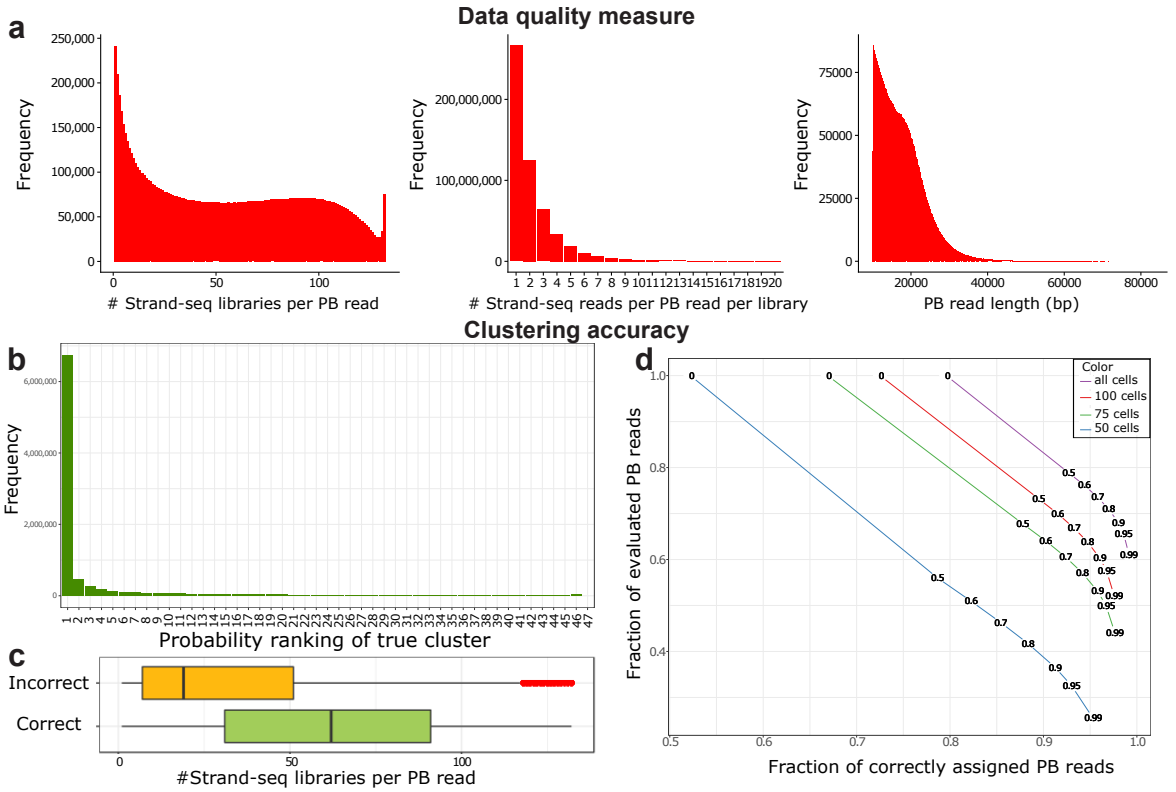


Figure 4.3: Data quality measures. a) Unfiltered data after mapping of Strand-seq reads to PacBio reads. Clustering accuracy is reported after filtering out 5% of PacBio reads with the highest Strand-seq read coverage. b) Distribution of PacBio reads based on the ranks of their true clusters sorted by probabilities (the cluster with the highest probability for any given PacBio read has rank 1 etc.) c) Distribution of the amount of Strand-seq libraries being represented per PacBio read as a function of a given PacBio read being assigned to a correct or incorrect cluster (chromosome and directionality). d) The accuracy for various probability thresholds (0, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95 and 0.99, written in blue circles) among PacBio reads represented in at least 5 Strand-seq libraries. Each curve represents a different number of Strand-seq libraries used (132, 100, 75, 50).

of the data, where each PacBio read is covered by only a handful of Strand-seq reads from a few libraries. This data sparsity is explained by observing the limited PacBio read length (Figure 4.3a, right) and the fact that Strand-seq is a single-cell sequencing technique with a limited coverage per library.

Clustering accuracy. For each PacBio read, we sorted the clusters in decreasing order based on their soft clustering probabilities. We then computed the rank of the true cluster in this sorted list for every PacBio read. The histogram in Figure 4.3b shows the distribution of these ranks across PacBio reads. For the majority (74.6%) of PacBio reads, the true cluster appeared at rank 1, meaning that the cluster with the highest probability was the true (correct) cluster. This means that such reads can be correctly clustered if we choose only the cluster with the highest probability. Besides that, there was a noticeable amount (11.5%) of PacBio reads with ranks 2-5,

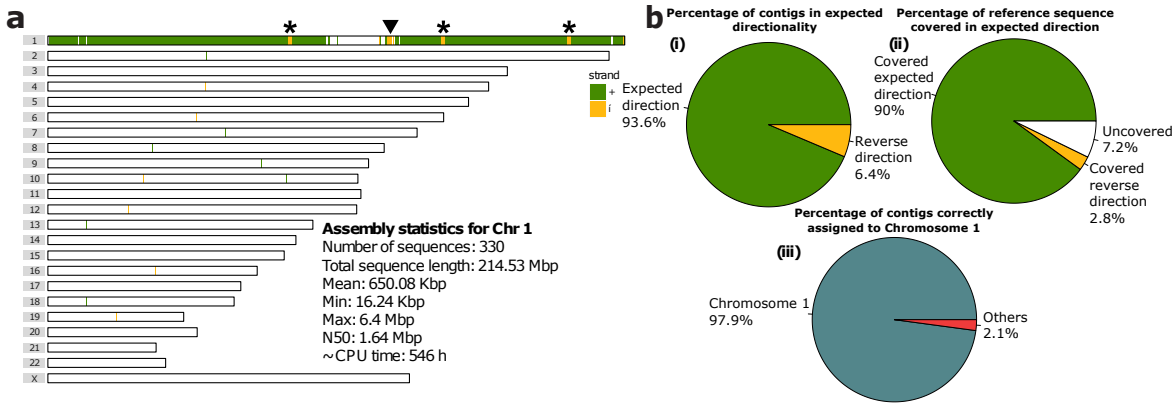


Figure 4.4: An ideogram showing genomic locations of contigs assembled from PacBio reads assigned to Chromosome 1 by SaaRclust. a) Horizontal lines represent individual chromosomes and vertical lines (green and yellow) depicts genomic location and directionality ('+': green, '-': yellow) of contigs mapped against referenge genome assembly. Black arrowhead points to a switch in contig directionality overlapping with known inversion. Asterisks points to directionality switches not presented as inversion before. Text inset presents various assembly statistics. CPU time is reported for the whole *de novo* assembly pipeline including read correction, read trimming and assembly using Canu. b) Statistics of Chromosome 1 assembly (i) Reports how many contigs were mapped in expected directionality to the reference genome. (ii) Shows total percentage of Chromosome 1 covered by contigs with both expected and reversed directionality. White chunk represents uncovered portions of Chromosome 1. (iii) Illustrates the fraction of assembled contigs mapped to Chromosome 1.

highlighting the benefits of soft clustering: In such a way, some PacBio reads can be assigned to a small list of clusters with a true one among them, even though there is an ambiguity with respect to the cluster assignment. These results are in line with the fact that some PacBio reads have a low Strand-seq coverage and true clusters might not be well distinguishable from the others.

To see how confidently we can assign each PacBio read to the chromosome with the highest probability, we computed the differences between the highest and second highest probability for all PacBio reads and observed that it was larger than 0.95 in 65.2% and smaller than 0.05 in 13% of all cases. This indicates that for the majority of PacBio reads, the difference is quite pronounced, whereas only a minority shows an ambiguous signal (those with sparse Strand-seq coverage), which is in line with the statistics displayed in Figure 4.3b.

Next, we sought to investigate the main determinants of a PacBio read being assigned to a correct cluster (chromosome). We assigned each PacBio read to the cluster with the highest probability. We evaluated each cluster assignment as correct or incorrect by comparison to the known original chromosome of each PacBio read. Subsequently, we investigated the distribution of the number of Strand-seq libraries being represented per PacBio read in the set of correctly and incorrectly clustered PacBio reads, respectively, as shown in Figure 4.3c. It is evident that there is a clear difference between the number of Strand-seq libraries represented in these two groups of

PacBio reads, with median values of 62 and 19 for correctly and incorrectly assigned PacBio reads, respectively. The low number of represented Strand-seq libraries in the incorrectly clustered PacBio reads meets our expectation that finding the true cluster is difficult when the data is too sparse. However, according to Figure 4.3b, the true cluster for these sparse data usually lies among the top clusters. The red points in the yellow box plot show the outliers that likely correspond to PacBio reads falling in repetitive regions of the genome and hence are receiving a lot of Strand-seq reads. Moreover, PacBio reads coming from repetitive regions of the genome are prone to have mis-mapped Strand-seq reads what might violate observed strand states.

For evaluating the accuracy of our clustering algorithm, we filtered out all PacBio reads that are represented in less than 5 Strand-seq libraries, which leads to a set of remaining reads with an average genome coverage of 48.9×. Among those selected PacBio reads, we computed the clustering accuracy using a set of probability thresholds (0, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95 and 0.99). In other words, we only evaluated PacBio reads whose maximum cluster probability was above the respective threshold (Figure 4.3d). Additionally, results for using varying numbers of Strand-seq libraries are represented as different curves, confirming the importance of including a large number of Strand-seq libraries. For all curves, there is a clear trade-off between the fraction of assigned reads and the clustering accuracy. That is, the more stringently we filter, the higher the accuracy, which confirms that the posterior probabilities indeed capture the degree of certainty about the assignment of a read to a chromosome. For instance, we were able to reach very high clustering accuracy (97.0%) corresponding to the probability threshold 0.8, while retaining 71.0% of PacBio reads. This amount of PacBio reads was sufficient to cover the human genome with 34.8× coverage. With a threshold of 0.99, we attain an accuracy of more than 99% while still retaining 61.1% of all reads and reaching a genome coverage of 30.1× (rightmost dot in top curve in Figure 4.3d).

Hard clustering-based versus random initialization. To assess the merits of our initialization procedure based on hard clustering, we replaced it by random initialization and compared the results. We ran the EM algorithm for the same number of iterations (50) and observed extremely poor performance for random initialization: When assigning each PacBio read to the cluster with maximum probability, we obtained an accuracy of 9.6% using random initialization as opposed to an accuracy of 79.6% when using the hard-clustering initialization (leftmost data point in top curve of Figure 4.3d). This experiment shows that the hard clustering step, which is orders of magnitude faster than the EM procedure (see Section 4.3.5), improves the final results drastically.

De novo assembly on clustered PacBio reads. Lastly we have tested performance of *de novo* assembly on clustered PacBio reads. We selected all PacBio reads that were represented in at least 5 single-cell libraries and that were assigned to Chromosome 1 with a probability of 0.5 and higher. This yields 489,203 of PacBio reads with a total sequence length corresponding to 35× coverage of Chromosome 1. Recall that the outcome of SaaRclust are two clusters, one contains PacBio reads in forward and the other contains PacBio reads in reverse direction. We have reverse complemented all PacBio reads assigned to reverse directionality cluster. This way we have synchronized directionality of all Pacbio reads belonging to Chromosome 1. The resulting reads were used as input to the Canu assembler [52] with parameters `correctedErrorRate=0.1` and `minOverlapLength=200`. This setting resulted in 460 contigs for Chromosome 1

with an N50 length of 1.05 Mbp as reported by Assemblytics [76].

Next, we explored whether the contiguity can be further improved by including more reads. To this end, we also included reads where Chromosome 1 was among the top clusters with probabilities markedly higher than the rest of them, based on detecting a peak in the differences between pairs of consecutive probabilities. With this approach, we have increased the number of PacBio reads assigned to Chromosome 1 to 667,346, corresponding to 47× coverage of Chromosome 1. We have repeated the assembly (using the same Canu parameters) with this new read set and obtained a more contiguous assembly consisting of 330 contigs (N50=1.64 Mbp) that cover 92.7% of Chromosome 1 after mapping them to the reference genome (Figure 4.4a,b). Interestingly, since all contigs are expected to be of the same directionality, any change in directionality might be an indication of structural variation. Based on this assumption we have been able to confirm a large inversion on Chromosome 1 previously reported by [94] (Figure 4.4a, arrowhead). However, we observed other regions of switched directionality that do not correspond to known inversions (Figure 4.4a, asterisks). Overall, the *de novo* assembly of pre-clustered PacBio reads provided highly specific contigs of which the vast majority (98%) was localized on the expected genomic chromosome (Figure 4.4b, iii). Moreover, almost all (93.6%) assembled contigs were mapped back to the reference sequence in an forward directionality covering 90% of Chromosome 1 (Figure 4.4b, i,ii).

4.5 Discussion

We presented a latent variable model and a corresponding EM algorithm to leverage single-cell strand sequencing data for clustering long sequencing reads by chromosomal origin and directionality. We implemented this algorithm in an R package, called SaaRclust, and tested it on the female human genome NA12878. SaaRclust exhibits a high accuracy even though the input data is extremely sparse and the read mapping process is complicated by the high error rates of PacBio CLR sequencing. It is the first tool able to cluster *long reads* by chromosome. This constitutes a major improvement compared to BAIT [43] and ContiBAIT, [81], which can perform clustering at the level of *contigs*, but are not designed to work with sparse read-level data.

We observed that the reliability of assigning a given PacBio read to a chromosome strongly depends on the Strand-seq coverage it received. In case of ambiguity, however, the true chromosome was among the top five clusters in most of these cases. The reliability of our clustering is further corroborated by the quality of the resulting *de novo* assembly of Chromosome 1, which achieved a high specificity of assembled contigs to Chromosome 1. Most likely, optimization of assembly parameters will bring further improvement in both accuracy and contiguity of assembled genomes using our approach and we plan to thoroughly test this on whole genomes in the future. Modifying existing assemblers to take advantage of the facts that the input reads have synchronized directionality and come with probabilities for chromosomal assignments is another promising direction we plan to explore.

As third generation sequencing technologies advance further, reads are anticipated to become even longer. We expect a major boost in the performance of SaaRclust on

longer reads, such as from the Oxford Nanopore Technologies platform. Beyond that, our single-cell sub-sampling analysis shows that increasing the number of single-cell Strand-seq libraries enhances the clustering accuracy significantly, and saturation has not yet been reached.

We hypothesize that PacBio reads with poor clustering accuracy despite sufficient coverage are originating from repeat regions in the genome. Such ambiguities can be accounted for by extending our model, and we plan to explore the potential of our approach for resolving segmental duplications in the future.

Here, we have focused on developing the necessary methodology and have benchmarked its performance. SaaRclust has been adjusted to cluster assembly contigs and successfully used in the core trio-free haplotype assembly pipeline [88], which is one of the first chromosome-scale haplotype assembly pipelines without parental data. This diploid haplotype assembly pipeline has been used in the second phase of Human Genome Structural Variation Consortium (HGSVC2) for assembly of 64 haplotype-resolved human genomes from different populations [23].

There is a wealth of other applications of our framework that can be addressed in the future. Highly rearranged genomes, like those resulting from somatic and germline chromothripsis events, are interesting cases for future studies since they are difficult to resolve with extant methods. Beyond human genomes, we envision our method to be of high utility for the assembly of plant genomes, which can be very large. To this end, extending the framework to higher ploidies, which are inherent to many plant genomes, would be an important avenue to pursue.

Chapter 5

Haploclust: clustering long DNA sequences by haplotype

This chapter presents clustering of assembled unitigs by their original chromosome and haplotype using the assembly graph structure. The aim of this project was doing haplotype bipartition of long DNA sequences as well as chromosome clustering. This study has been conducted under supervision of Tobias Marschall. The adjustment and implementation of StrandPhaseR algorithm has been done by David Porubsky. A part of the conceptual figure 5.1 including the overlap graph and their bubble structures has been initially designed and created by David Porubsky, and later I added more illustration and adjusted the figure for this thesis. The rest of the work including designing and implementation of the computational workflow and the rest of the figures were my own contribution. I would like to thank Mikko Rautiainen for his help and fruitful discussions on the assembly graph data structures and Fawaz Dabbaghieh for providing me with his bubble detection tool.

5.1 Introduction

Haplotype phasing is essential for a full reconstruction of diploid genomes, such as human genomes. Many haplotype phasing tools rely on the existence of a reference genome to create a bipartition of the sequencing reads into haplotypes, and reconstruct the haplotypes [5, 84]. Because of existence of complex genomic rearrangements, it is indeed useful and practical to remove the reference bias inherent in resequencing approaches and directly phase the haplotypes from the sequencing reads. Moreover, a reference genome is still not available for all species. There have been some studies aiming to phase and assemble diploid genomes without using a reference. Many diploid assemblers [14, 30, 113] are based on local haplotype separation, and they cannot phase haplotypes spanning the whole chromosomes because of lacking global haplotype information. Trio binning approaches [53] require parents sequencing data to phase the genome of an individual, which is not always available.

Integration of Hi-C or Strand-seq data has been proved to be successful in combination with third generation long read sequencing in order to reach high contiguity of haplotype assemblies. Hi-C and strand sequencing technologies have long range phas-

ing information that can be integrated with low sparse haplotype phasing information from third generation sequencing reads leading to long contiguous haplotypes. This data integration has been used in two recent assembly pipelines [88] and [31] that resulted in long contiguous haplotigs in the absense of trio data. These two assembly workflows are based on making a draft reference genome by performing an initial assembly of reads, then doing haplotype phasing and the final round of assembly on the set of haplotype-partitioned contigs, which can entail a high computational burden.

The computational burden of this first round of assembly and performing reference-based alignment, variant calling and haplotype phasing could be avoided by exploiting the structure of the overlap graph for haplotype phasing. The concept of graph-based haplotype phasing has been previously used [30]; however this approach was based on making local connections between heterozygous parts of the graph using PacBio reads, which does not result in long contiguous assemblies. The contiguity problem could be resolved by integration of Hi-C or Strand-seq data in graph-based phasing. The new version of the Hifiasm assembler uses the bubble structure of heterozygous graph parts and phases them using Hi-C data integration [12]. It produces long contiguous haplotigs, however phasing the contigs across the entire chromosomes is not feasible in Hifiasm using Hi-C integration.

Strand-seq data has a sparse chromosome-wide haplotype information, which is suited for whole-chromosome haplotype assembly. The novelty of this work is the integration of Strand-seq into an overlap graph to cluster the set of unitigs into their chromosomes and haplotypes, which can result in chromosome-wide haplotype assignment to unitigs and contigs. Our pipeline is the first graph-based pipeline that uses Strand-seq data for a chromosome-wide haplotype phasing of the overlap graph, which can result in highly contiguous haplotype assemblies.

5.2 Haploclust idea

As discussed in the previous chapter, Strand-seq data can be used to cluster long sequencing reads into their original chromosome and direction. In addition to the chromosome-specific signal, single cells of type WC (the cells inherited one Watson strand from one parent and one Crick strand from the other parent) are informative of haplotypes in each chromosome. In this work, we show how to utilize the haplotype information of Strand-seq data to extend our chromosome-clustering algorithm to haplotype clustering.

Strand-seq has been successfully used in chromosome-wide haplotype phasing [87] in a reference-based setting. A core algorithmic part of the aforementioned study is the StrandPhaseR algorithm [87], which uses the Strand-seq data on the set of shared heterozygous SNV positions to build consensus haplotypes (see Section 5.2). It is a greedy algorithm that uses this shared SNV loci information to assign W and C strands to their corresponding haplotype in each chromosome.

In the *de novo* case, the reference genome and hence the set of heterozygous SNV loci are absent. However, the heterozygous genomic loci can be detected from the assembly graph. Divergence of DNA sequences in the heterozygous parts of the genome result in bubble structures in the overlap graph (see Figure 5.1c). For haplotype sorting

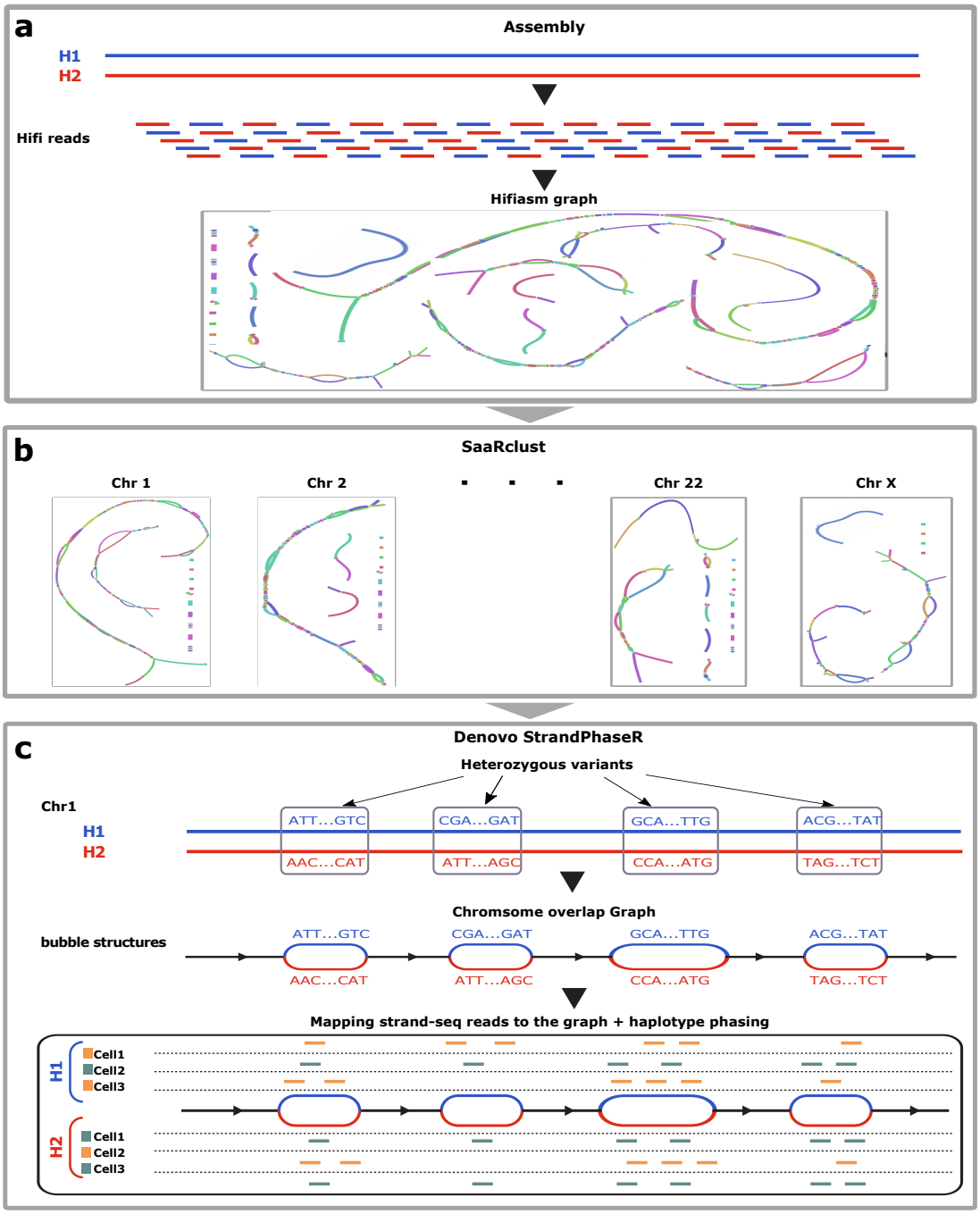


Figure 5.1: Haploclust overview. a) Assembly of HiFi reads using Hifiasm tool. It outputs an overlap graph. b) Clustering overlap graph unitigs using Strand-seq data by SaaRclust tool. c) *De novo* StrandPhaseR on every chromosome-specific overlap graph. Heterozygous variants create bubble structures in the overlap graph. The number of maximal unique matches of Strand-seq reads in bubbles is inputted to *de novo* StrandPhaseR resulting in a set of phased bubbles and single-cell strand phased states. The phased strand states are used afterwards to phase the whole set of unitigs.

of W and C strand states in the absence of a reference genome, we explore these structures in the assembly overlap graph to generalize the StrandPhaseR algorithm to the *de novo* setting.

In the overlap graph, haplotype alleles are encoded in the two divergent paths of bubbles. We refer to these haplotype alleles as *bubble alleles*. Strand-seq reads from WC single cells have their haplotype-specific signal on the bubbles of the overlap graph in each chromosome, that is, in heterozygous parts of the bubble allele sequences, the reads from W and C strands cover the bubble alleles of two different haplotypes in each of these single cells (see Figure 5.1c).

The main idea of our pipeline is to perform the Strand-seq based chromosome clustering (SaaRclust [37]) and haplotype phasing (StrandPhaseR [87]) in the graph space. We chose HiFi sequencing technology because HiFi CCS reads are highly accurate third generation sequencing reads taking advantages of having long read length and low error rates. For assembling HiFi reads, we use Hifiasm [13], which is a fast overlap graph-based assembler for HiFi reads that enabled us to study and analyze the heterozygous bubble structures of diploid genomes.

Figure 5.1 shows an overview of this haplotype clustering pipeline. We first assemble the HiFi reads using Hifiasm [13] and then split the assembled unitigs and assembly graphs by their chromosome using SaaRclust [37]. After Hifiasm graph construction, we detect bubbles in the overlap graph and then create tables of Strand-seq W/C read counts by finding the Strand-seq maximal unique exact matches in the set of bubbles of the chromosome-split overlap graphs. We later perform a *de novo* version of StrandPhaseR to phase bubble alleles and sort WC strand states by their haplotypes. *De novo* StrandPhaseR is our adjusted StrandPhaseR algorithm to use graph-based bubble alleles instead of reference-based SNV alleles. At the end, we phase the set of all Hifiasm unitigs that have maximal unique Strand-seq matches from haplotype-clustered Strand-seq reads. The final output of our pipeline is a clustering of Strand-seq reads and Hifiasm unitigs by their original chromosome and haplotype, which can be used for haplotype assembly of diploid genomes. The details of the pipeline are elaborated in the following section.

5.3 Haploclust computational pipeline

We developed a pipeline that performs assembly, chromosome clustering, bubble detection, and haplotype phasing of the overlap graph using our *de novo* versions of SaaRclust and StrandPhaseR adjusted to the graph space.

5.3.1 Input data

Our pipeline gets as input a Fasta/Fastq file containing long accurate HiFi reads, and a set of Fasta/Fastq files including Strand-seq reads per single cell. Note that our pipeline can run on any overlap graph that can be constructed from other sequencing reads. One could easily skip the assembly step in our pipeline and input an externally-created overlap graph into the pipeline.

5.3.2 Merging the overlapping pairs of Strand-seq reads

In the paired-end mode, the read pairs produced from Strand-seq technology are usually overlapping. We merge the overlapping pairs of Strand-seq reads by PEAR [124]. We use the default minimum overlap size in PEAR (10 bp) for read pair merging. For the Strand-seq reads whose pairs do not have this minimum length of overlap, we use only the first read of a pair. For the rest of the reads that are successfully merged, we use the merged pairs as the input for downstream steps. This processing step enables us to increase the Strand-seq read length, which can be useful for the alignment, clustering and phasing tasks.

5.3.3 Assembling long reads

We use Hifiasm [13] for the whole genome assembly of input HiFi reads. We use the raw unitigs output graph of Hifiasm for further processing. The raw unitigs graph is a less processed output graph of Hifiasm in which the sequences of the two haplotypes are not collapsed yet. Hence, the information of the heterozygous sequences exist in the bubble structure of the raw unitigs graph, which we use later on for haplotype phasing.

5.3.4 Chromosome clustering and strand state detection

We align Strand-seq reads to the set of Hifiasm raw unitigs by BWA [59] alignment tool, and cluster the unitigs by chromosome and direction using SaaRclust (Chapter 4).

Using the clustering information of unitigs and the alignments of Strand-seq reads to unitigs, we cluster the Strand-seq reads by chromosomes and directions. Note that SaaRclust is able to pair forward and backward clusters that come from the same chromosome (see Section 4.2.2).

Every unitig aligned to a single-cell Strand-seq read implies a clustering for the read depending on the alignment direction. If the mapping direction is forward, the unitig's cluster is supported for the Strand-seq read, otherwise the chromosome pair of the unitig's cluster is supported. More precisely, let us denote $\overrightarrow{\mathcal{A}}(S)$ and $\overleftarrow{\mathcal{A}}(S)$ as the sets of unitigs to which the single-cell Strand-seq read S aligned in forward and backward directions, respectively. Let us denote the cluster of a unitig U as $\mathcal{C}(U)$, and the chromosome pair of a cluster $\mathcal{C}$ as $\mathcal{C}_p$. We define the represented clusters set $\mathcal{CS}(S)$ for the Strand-seq read S as follows:

$$\mathcal{CS}(S) = \{\mathcal{C}(U) \text{ for } U \text{ in } \overrightarrow{\mathcal{A}}(S)\} \cup \{\mathcal{C}_p(U) \text{ for } U \text{ in } \overleftarrow{\mathcal{A}}(S)\}.$$

If the represented clusters set has a unique cluster for a Strand-seq read, we assign that cluster to the Strand-seq read, otherwise we don't cluster the read because of ambiguity. As a result of the clustering, for each chromosome, we have two clusters of unitigs/Strand-seq reads in forward and backward directions.

We also detect single-cell strand states per chromosome using the strand state probabilities, which is an output of SaaRclust. After clustering, we split the set of

unitigs, Strand-seq reads, and the Hifiasm graph into separate chromosomes. From this step on, the rest of the data analysis is conducted per chromosome separately.

5.3.5 Bubble detection

We detect bubbles in the Hifiasm graph of each chromosome using the BubbleGun [17] tool. BubbleGun is a fast tool implementing the algorithm from [82]. The set of simple bubble structures detected by BubbleGun give us candidate heterozygous unitigs that we later use for haplotype phasing.

5.3.6 Strand-seq maximal unique matches

For each chromosome, we detect maximal unique exact matches of Strand-seq reads in every unitig using the BWA fastmap [59] tool, which is a fast tool for detecting exact sequence matches. The unique matches of Strand-seq sequences in bubbles are used for haplotype phasing of the graph and unitigs in the following step.

Note that for the set of Strand-seq reads that cover a heterozygous site, the maximal unique match should happen with the unitig of the same haplotype allele. The reads that cover a fully homozygous sequence will be removed from the downstream haplotype phasing step. Theses reads either overlap with a bubble structure with two (non-unique) maximal sequence matches, or they overlap with non-bubble parts of the graph, which will not be used in the haplotype phasing analysis of the pipeline. Therefore, this step enables us to detect haplotype-informative Strand-seq reads and correspond them to the right bubble allele in the bubble structures in the graph.

5.3.7 *De novo* StrandPhaseR

The aforementioned unique exact matches provide us with the information on the number of W and C Strand-seq reads that cover bubble alleles from each single cell. For each chromosome, we create two matrices (one matrix for each cluster in the chromosome clusters pair) including the coverage of bubble alleles by WC Strand-seq single cells. The rows of matrices correspond to single cells and columns correspond to bubbles (Figure 5.2).

The elements of these matrices can be one of $\{-, 0, 1, 2\}$ values indicating that the bubble is not covered by a single cell, or the covered bubble allele is 0 (first allele), 1 (second allele), or 2 (mixed), respectively. We consider a bubble coverage with mixed alleles if the fraction of reads covering the first allele is between 0.25 and 0.75, otherwise the bubble allele with the majority of Strand-seq matches is selected as the covered allele. More precisely, let us define m_k as the coverage matrix for cluster k and $C_{j,k,b,al}$ as the number of unique matches of Strand-seq reads coming from single cell j and cluster k that cover allele $al \in \{0, 1\}$ of bubble b. We define the bubble allele coverage matrix as below:

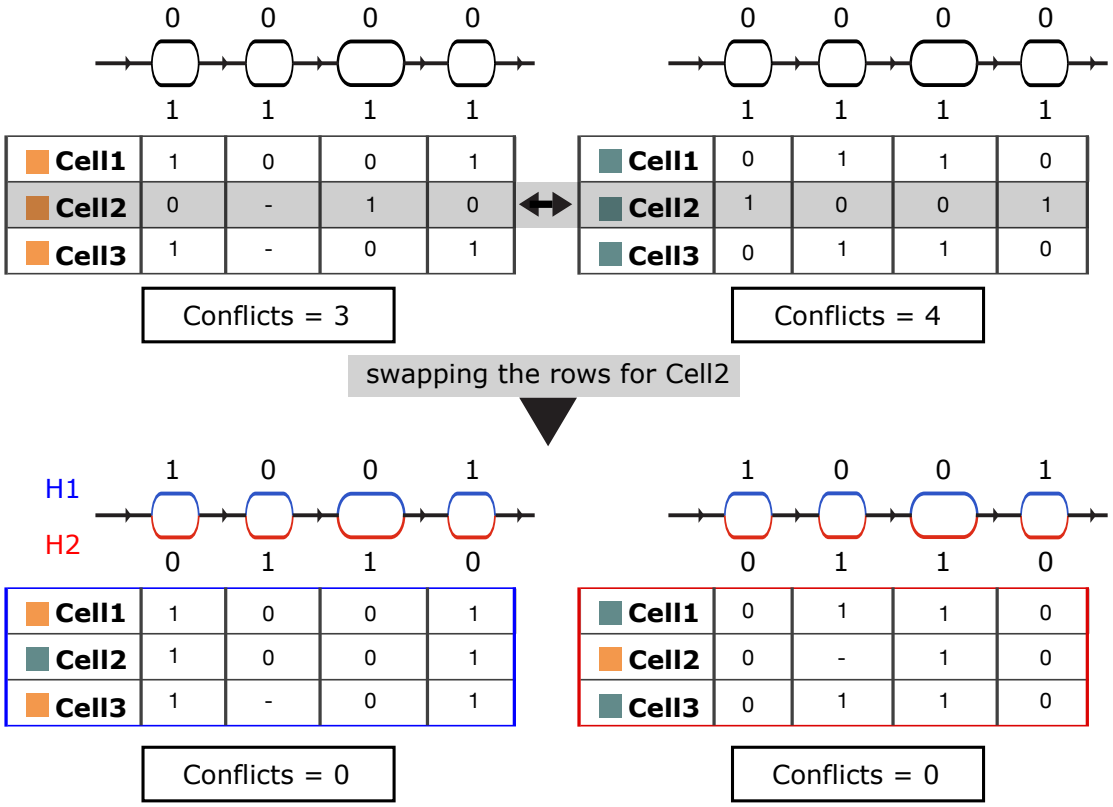


Figure 5.2: *De novo* StrandPhaseR algorithm. Each chromosome has two matrices representing coverage of bubble (columns) alleles in WC single cells (rows). Initially, there is one matrix for W single cells (top left) and one matrix for C single cells (top right). StrandPhaseR aims to minimize the number of disagreements (conflicts) across the columns by swapping the single-cell rows iteratively. In this example, swapping the row for single cell 2 results in two matrices with zero conflicts (bottom).

$$m_k[j,b] = \begin{cases} - & \text{if } C_{j,k,b,0} + C_{j,k,b,1} = 0 \\ 0 & \text{if } \frac{C_{j,k,b,0}}{C_{j,k,b,0}+C_{j,k,b,1}} > 0.75 \\ 1 & \text{if } \frac{C_{j,k,b,0}}{C_{j,k,b,0}+C_{j,k,b,1}} < 0.25 \\ 2 & \text{if } 0.25 \leq \frac{C_{j,k,b,0}}{C_{j,k,b,0}+C_{j,k,b,1}} \leq 0.75 \end{cases}$$

It results in two separate coverage tables for forward and backward Strand-seq reads for each chromosome. Each table has the number of Strand-seq matches for every single cell and bubble.

Figure 5.2 shows the bubble coverage matrices for three single cells corresponding to the example chromosome at Figure 5.1c. Initially, there is one bubble coverage matrix for the forward Strand-seq reads and one matrix for the backward Strand-seq reads (Figure 5.2 top). *De novo* StrandPhaseR is a greedy algorithm that iteratively swaps the single-cell pairs of rows until it reaches two matrices corresponding to the two haplotypes with the minimum number of conflicts across the columns (Figure 5.2

bottom). This algorithm results in a set of phased bubbles and phased strand states for WC single cells for each chromosome.

5.3.8 Clustering of unitigs by haplotypes

The result of *de novo* StrandPhaseR is a set of haplotype-phased Strand-seq reads in single cells of type WC per chromosome. Note that we can phase all Strand-seq reads that come from a single cell with a phased WC strand state, regardless of whether the read overlaps with a bubble structure or not. This phasing information of WC Strand-seq reads provides haplotype-specific signals in all heterozygous unitigs that have unique matches of phased Strand-seq reads.

We exploit this phasing information and the unique matches of Strand-seq reads in unitigs to phase the set of unitigs into their original haplotypes. To this aim, we compute the number of unique Strand-seq matches that are assigned to H1 and H2 haplotypes in each unitig. Let us define the two aforementioned quantities as N_{H1} and N_{H2}. If the fraction $\frac{N_{H1}}{N_{H1}+N_{H2}}$ is between 0.25 and 0.75 for a unitig, we classify the unitig as ambiguous haplotype because there is a mixture of H1 and H2 signals, which can happen because of potential misassemblies, errors in alignment or chromosome clustering. Otherwise, we assign the unitig to the haplotype with highest unique Strand-seq count, i.e., we assign it to H1 if $N_{H1} > N_{H2}$ and to H2 if $N_{H2} > N_{H1}$.

5.4 Results

We evaluated the performance of our pipeline on HG00733, which is one of the human genome samples of the HGSVC project. We used human reference genome hg38 and phased VCF files resulting from trio-based haplotype phasing of HG00733 [10] as ground truth for benchmarking our haplotype clustering pipeline. Note that we use the reference genome only for evaluation purpose, and the proposed pipeline does not use any reference genome.

Our Hifiasm assembly experiment produced $74,415$ unitigs, 97% of which were mapped to a known chromosome in the reference genome. The accuracy of SaaRclust chromosome clustering was 97.1% among these set of unitigs. The number of unitigs that were haplotagged with the reference-based trio haplotype phasing was 48878 ($\approx$ 65.7% of unitigs).

Figure 5.3a shows the Venn diagram of the set of haplotagged unitigs (based on ground truth haplotypes) and the set of unitigs with haplotype assignment from Haploclust. Among the set of heterozygous unitigs that were haplotagged in the ground true data, the Haploclust pipeline clustered 66.3% unitigs with 97.8% accuracy. There are also a small fraction (3.6%) of not-haplotagged (ground truth) unitigs that are assigned to a haplotype in Haploclust, which can be either false or potentially newly discovered haplotype assignments.

Figure 5.3b shows that in terms of sequence base pair, 9.8% of the total unitig length come from the homozygous unitigs, and 76.6% of the unitig length come from the heterozygous unitigs that were correctly clustered by chromosome and haplotype in the Haploclust pipeline. A fraction of 8% of the unitig length correspond to the unitigs

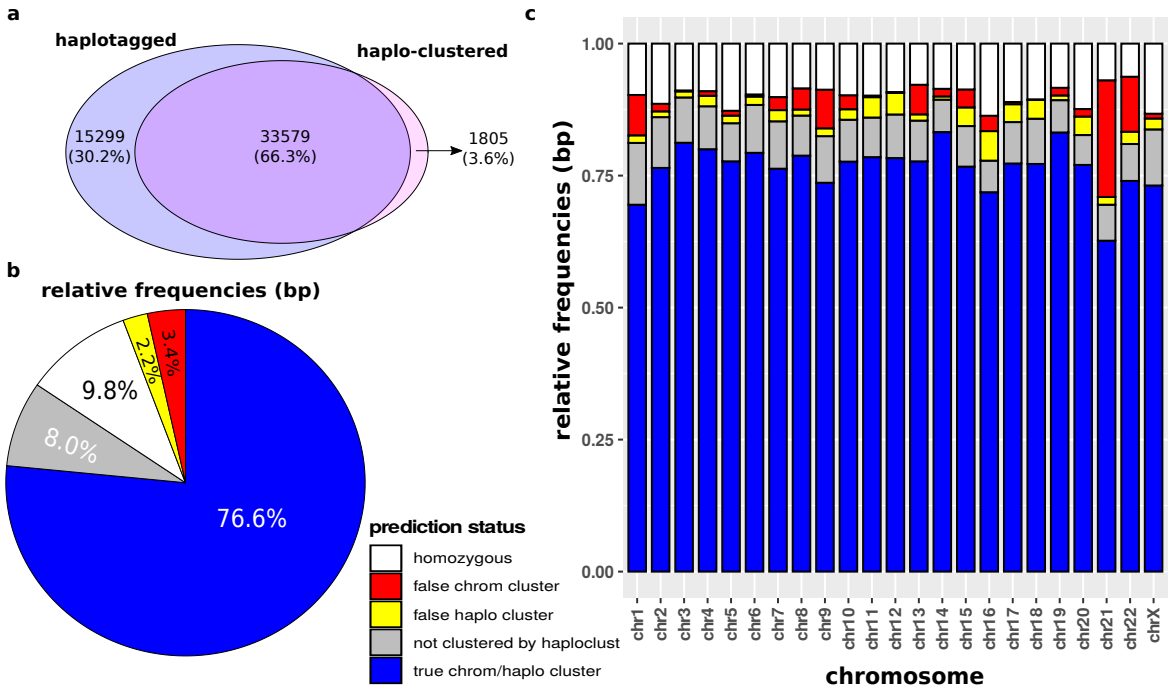


Figure 5.3: Haploclust results on HG00733. a) Venn diagram of the set of haplotagged unitigs (based on benchmark data) and haplotype-clustered unitigs. b) Pie chart of the relative frequencies of the unitigs falling into five different prediction categories: "false chromosome cluster", "false haplotype cluster", "homozygous", and "true chromosome and haplotype cluster". c) Bar plot representation of the same relative frequencies per chromosome.

that are correctly assigned to their chromosome by SaaRclust, but not clustered by their haplotypes. These are the set of unitigs that are relatively small and do not have enough number of Strand-seq matches to be clustered by haplotype. There are small fractions of unitigs that were assigned to a wrong chromosome or a wrong haplotype, constituting 3.4% and 2.2% of the total unitig base pairs, respectively. Next section will represent detailed examples of different haplotype assignments in Hifiasm unitigs.

5.4.1 Phased Hifiasm graph patterns

We observed several patterns that were common in the Hifiasm graph in all chromosomes over the whole genome. Figure 5.4 shows examples of four different common patterns in the phased Hifiasm graph of chr22. As shown in the examples, the colors (predicted haplotypes) and the labels (ground true haplotypes) match for all colored graph nodes, which means that the haplotype clustering was accurate for the set of haplotype-clustered unitigs.

The first example (a) shows a proper haplotype clustering even on a *broken bubble*. A *broken bubble* is similar to simple bubble structure with the difference that one of the bubble alleles is disconnected alike Figure 5.4a. More precisely, a broken bubble is a subgraph induced from five vertices a, b, c, d, e with edges $(a, b), (c, e), (a, d), (d, e)$.

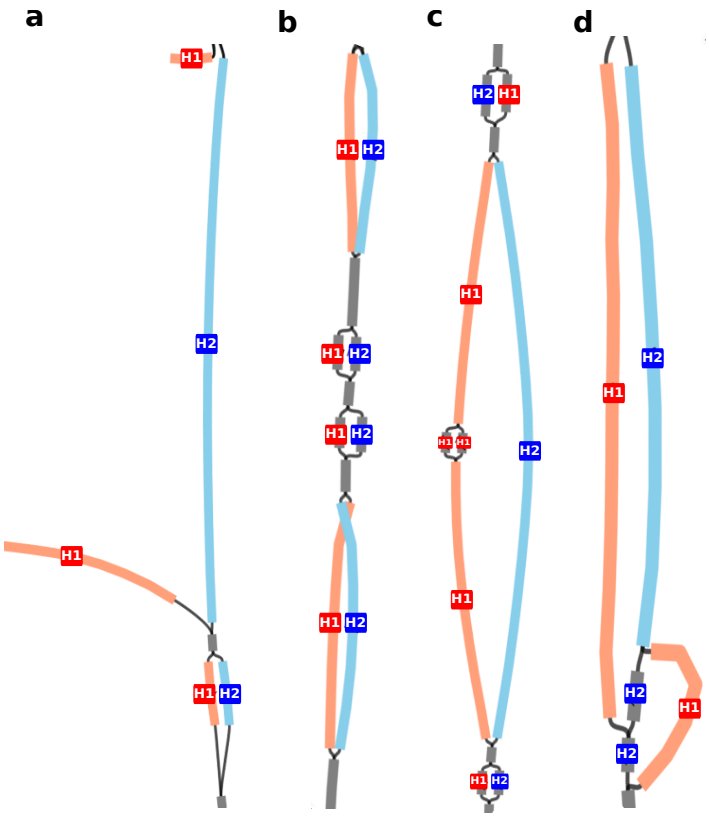


Figure 5.4: Examples of Phased graph. Examples of four common patterns in Hifiasm graph of chr22. Node colors show the predicted haplotypes (orange:H1, blue:H2), and labels are the ground true haplotypes. a) A complete accurate haplotype clustering even on a broken bubble. b) Two large haplotype-assigned bubbles and two smaller heterozygous bubbles with no haplotype assignments. c) Example of nested bubbles. The large unitigs in the nested bubble are correctly haplotype clustered. The small bubble inside the other bubble is a result of sequencing errors and is not assigned to a haplotype. d) A misassembly of H1 and H2 reads resulting in a mixture of paths of the two haplotypes. The large unitigs are still assigned to their correct haplotypes.

Note that a broken bubble is not detected by BubbleGun and hence not used in StrandPhaseR. However, it is successfully phased in our pipeline in the downstream haplotype phasing step after having the simple bubbles and Strand-seq reads phased by StrandPhaseR.

The second example (b) represents two successfully haplotype-assigned and two unassigned bubbles. The bubbles that are not haplotype-assigned are relatively small bubbles, which usually do not have enough unique Strand-seq matches and cannot be phased by Strand-seq data. These small unassigned bubbles can be seen in the third and fourth examples as well.

The third example (c) shows a nested bubble, which has another smaller bubble inside one of the bubble alleles. These structures can be created because of sequencing or assembly errors. Similar to the first example, these types of bubbles are also not

used as a proper heterozygous bubble in our pipeline, but the main parts of the bubble (excluding the small bubble inside the nested bubble) are successfully assigned to their right haplotype one step after StrandPhaseR.

The forth graph (d) is an example of a misassembly in the Hifiasm graph, where the paths of the two haplotypes are mixed together, but Haploclust has successfully assigned the long unitigs to their correct haplotypes. This example shows a potential capability of Haploclust to be applied to correct some haplotype misassemblies.

5.4.2 Discussion

We proposed Haploclust, a pipeline that builds a Hifiasm overlap graph and clusters the unitigs by their original chromosome and haplotype using the combination of HiFi and Strand-seq reads. It is the first graph-based pipeline developed for *de novo* and trio-free chromosome and haplotype clustering using the combination of Strand-seq and HiFi reads. Our pipeline can be applied not only to Hifiasm graph, but also to any overlap graph built from long reads.

Haploclust facilitates trio-free *de novo* diploid genome assembly by separation of haplotypes directly on the graph data structure, which is a more intuitive way with less computational burden compared to the previous *de novo* phasing pipeline [88]. It can separate the overlap graphs of the two haplotypes per chromosome. It can be further developed to assemble the haplotype-specific graphs to reconstruct the two haplotype assemblies per chromosome.

Based on the observed patterns in our phased overlap graphs, Haploclust can correctly phase large unitigs and results in chromosome-wide true haplotype assignments for large unitigs. The limitation of our method comes mainly from sparsity of Strand-seq data that leaves small unitigs unphased. We expect that integrating our method with the graph phasing based on Nanopore reads would solve the small not-phased unitigs. Strand-seq reads have sparse chromosome-wide haplotype information, and Nanopore reads have dense local haplotype signals, therefore the combination of both graph-based tools can lead to a more complete and contiguous graph-based haplotype assembly. In this direction, a future plan is to run Haploclust on the Verkko assembly graph, which is an assembly based on combination of HiFi and Nanopore reads, from the T2T project [80].

We observed some haplotype misassembly cases in the Hifiasm graph in which the unitigs can be still correctly phased using our pipeline. These examples show the potential of Haploclust to correct haplotype misassemblies and lead to more accurate assemblies.

In summary, Haploclust has been successful in phasing most of the overlap graph unitigs. In the future, we can develop our own graph traversal algorithm for assembling haplotype-specific overlap graphs. The integration of other sequencing data and tools enables us to phase larger fractions of unitigs that leads to more complete haplotype assemblies. It can be also generalized to polyploid graph-based phasing with notable applications in the field of plant and cancer genomics.

Chapter 6

Conclusion

This thesis consists of three main projects of my PhD studies. They are centered on different applications of single-cell Strand sequencing in genomic structural variation detection in single cells and *de novo* chromosome and haplotype clustering.

We developed scTRIP, a computational pipeline for discovery of structural variants in single cells. Our developed pipeline is able to detect a wide range of somatic SV classes in single cells, including deletion, duplication, inversion, and more complex SVs such as inverted duplication. scTRIP has been also successful in spotting breakage fusion bridge (BFB) cycles. Chromosomal rearrangements are a major source of cancer, and BFB is an example of these genome instabilities. We were able to detect and cluster heterogeneous BFB events in RPE-C7 single cells. Moreover, our analysis of T-ALL patient-derived sample showed that there might be a utility of scTRIP in the future for disease prognosis involving complex chromothripsis events.

scTRIP has been used for inversion detection as part of HGSVC2 project, where it was adjusted to use the SV breakpoints input based on orthogonal sequencing technologies [23] allowing for better breakpoint resolution.

The second project was SaaRclust, a tool for clustering long DNA sequencing reads into their original chromosomes and directions. We developed a latent variable model and an EM algorithm for performing the clustering task. It was the first tool enabling clustering of long sequencing reads by chromosome resulting in improvement in genome assembly. It has been successfully used as one of the core parts in a big genome assembly project [88], whose assembly workflow was later used for HGSVC2 [23] project. There can be different other applications of SaaRclust that can be explored in the future. One interesting future direction can be exploiting the potential of SaaRclust to discover chromosomal rearrangements such as chromothripsis events.

Lastly, we presented the Haploclust pipeline for haplotype clustering of overlap graph unitigs as well as chromosome clustering. The current pipeline works on the combination of Hifi and single-cell strand sequencing reads. The results on HG00733 show accurate haplotype clustering in long enough unitigs; however it cannot assign a haplotypes to short unitigs due to lack of exact Strand-seq matches. The explored graph structure examples indicate that Haploclust can successfully assign the correct haplotype even in more complex graph parts, including broken bubbles, and mis-assembled contigs. It shows the potential of Haploclust to correct haplotype mis-assemblies in the overlap graph. Haploclust can be further developed to produce the set of phased

assembled contigs by implementing our own graph traversal algorithm on the set of haplotype-split overlap graphs. The integration of Nanopore data into the graph-based phasing problem can improve the haplotype clustering, specially in shorter unitigs, and lead to more contiguous assembled contigs. A future direction is to run Haploclust on the Verkko assembly graph, which is a graph assembled from the combination of HiFi and Nanopore reads.

An interesting generalization of all of the presented methods is to adjust them to work on polyploid genomes, which is of high utility in cancer and plant genomics. In summary, we showed that single-cell strand sequencing technology has application in SV detection and genome assembly owing to its strand-specific chromosome and haplotype signals. The presented tools and pipelines have been successfully applied in big assembly and structural variant detection projects and can be further explored in other biological applications.

Bibliography

[1] C. Alkan, B. P. Coe, and E. E. Eichler. Genome structural variation discovery and genotyping. *Nature Reviews Genetics*, 12(5):363–376, 2011.

[2] M. Aubry, M. de Tayrac, A. Etcheverry, A. Clavreul, S. Saikali, P. Menei, and J. Mosser. Correction: From the core to beyond the margin: a genomic picture of glioblastoma intratumor heterogeneity. *Oncotarget*, 7(41):67685, 2016.

[3] S. F. Bakhoum, W. T. Silkworth, I. K. Nardi, J. M. Nicholson, D. A. Compton, and D. Cimini. The mitotic origin of chromosomal instability. *Current Biology*, 24(4):R148–R149, 2014.

[4] B. Bakker, A. Taudt, M. E. Belderbos, D. Porubsky, D. C. Spierings, T. V. de Jong, N. Halsema, H. G. Kazemier, K. Hoekstra-Wakker, A. Bradley, et al. Single-cell sequencing reveals karyotype heterogeneity in murine and human malignancies. *Genome biology*, 17(1):1–15, 2016.

[5] V. Bansal and V. Bafna. Hapcut: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics*, 24(16):i153–i159, 2008.

[6] D. R. Bentley, S. Balasubramanian, H. P. Swerdlow, G. P. Smith, J. Milton, C. G. Brown, K. P. Hall, D. J. Evers, C. L. Barnes, H. R. Bignell, et al. Accurate whole human genome sequencing using reversible terminator chemistry. *nature*, 456(7218):53–59, 2008.

[7] M. F. Berger, M. S. Lawrence, F. Demichelis, Y. Drier, K. Cibulskis, A. Y. Sivachenko, A. Sboner, R. Esgueva, D. Pflueger, C. Sougnez, et al. The genomic complexity of primary human prostate cancer. *Nature*, 470(7333):214–220, 2011.

[8] I. M. Campbell, C. A. Shaw, P. Stankiewicz, and J. R. Lupski. Somatic mosaicism: implications for disease and transmission genetics. *Trends in Genetics*, 31(7):382–392, 2015.

[9] P. J. Campbell, S. Yachida, L. J. Mudie, P. J. Stephens, E. D. Pleasance, L. A. Stebbings, L. A. Morsberger, C. Latimer, S. McLaren, M.-L. Lin, et al. The patterns and dynamics of genomic instability in metastatic pancreatic cancer. *Nature*, 467(7319):1109–1113, 2010.

[10] M. J. Chaisson, A. D. Sanders, X. Zhao, A. Malhotra, D. Porubsky, T. Rausch, E. J. Gardner, O. L. Rodriguez, L. Guo, R. L. Collins, et al. Multi-platform

discovery of haplotype-resolved structural variation in human genomes. *Nature communications*, 10(1):1–16, 2019.

[11] E. Check Hayden. Genome sequencing: the third generation, 2009.

[12] H. Cheng, G. T. Concepcion, X. Feng, H. Zhang, and H. Li. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nature Methods*, 18(2):170–175, 2021.

[13] H. Cheng, G. T. Concepcion, X. Feng, H. Zhang, and H. Li. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nature Methods*, 18(2):170–175, 2021.

[14] C.-S. Chin, P. Peluso, F. J. Sedlazeck, M. Nattestad, G. T. Concepcion, A. Clum, C. Dunn, R. O'Malley, R. Figueroa-Balderas, A. Morales-Cruz, et al. Phased diploid genome assembly with single-molecule real-time sequencing. *Nature methods*, 13(12):1050–1054, 2016.

[15] G. Ciriello, M. L. Miller, B. A. Aksoy, Y. Senbabaoglu, N. Schultz, and C. Sander. Emerging landscape of oncogenic signatures across human cancers. *Nature genetics*, 45(10):1127–1133, 2013.

[16] . G. P. Consortium et al. A global reference for human genetic variation. *Nature*, 526(7571):68, 2015.

[17] F. Dabbaghie, J. Ebler, and T. Marschall. Bubblegun: enumerating bubbles and superbubbles in genome graphs. *Bioinformatics*, 38(17):4217–4219, 2022.

[18] C. Darwin. On the origin of species, 1859, 2016.

[19] A. Davis, R. Gao, and N. Navin. Tumor evolution: Linear, branching, neutral or punctuated? *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1867 (2):151–161, 2017.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[21] M. A. DePristo, E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. A. Philippakis, G. Del Angel, M. A. Rivas, M. Hanna, et al. A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics*, 43(5):491, 2011.

[22] Y. Dou, H. D. Gold, L. J. Luquette, and P. J. Park. Detecting somatic mutations in normal cells. *Trends in Genetics*, 34(7):545–557, 2018.

[23] P. Ebert, P. A. Audano, Q. Zhu, B. Rodriguez-Martin, D. Porubsky, M. J. Bonder, A. Sulovari, J. Ebler, W. Zhou, R. S. Mari, et al. Haplotype-resolved diverse human genomes and integrated analysis of structural variation. *Science*, 372 (6537), 2021.

[24] N. Editorial. Method of the year 2013. *Nat Methods*, 11:1, 2014.

[25] R. Ekblom, L. Smeds, and H. Ellegren. Patterns of sequencing coverage bias revealed by ultra-deep sequencing of vertebrate mitochondria. *BMC genomics*, 15(1):1–9, 2014.

[26] E. Falconer, M. Hills, U. Naumann, S. S. Poon, E. A. Chavez, A. D. Sanders, Y. Zhao, M. Hirst, and P. M. Lansdorp. Dna template strand sequencing of single-cells maps genomic rearrangements at high resolution. *Nature methods*, 9 (11):1107–1112, 2012.

[27] E. R. Fearon and B. Vogelstein. A genetic model for colorectal tumorigenesis. *cell*, 61(5):759–767, 1990.

[28] L. A. Forsberg, D. Gisselsson, and J. P. Dumanski. Mosaicism in health and disease—clones picking up speed. *Nature Reviews Genetics*, 18(2):128, 2017.

[29] L. A. Forsberg, D. Gisselsson, and J. P. Dumanski. Mosaicism in health and disease—clones picking up speed. *Nature Reviews Genetics*, 18(2):128, 2017.

[30] S. Garg, M. Rautiainen, A. M. Novak, E. Garrison, R. Durbin, and T. Marschall. A graph-based approach to diploid genome assembly. *Bioinformatics*, 34(13): i105–i114, 2018.

[31] S. Garg, A. Fungtammasan, A. Carroll, M. Chou, A. Schmitt, X. Zhou, S. Mac, P. Peluso, E. Hatas, J. Ghurye, et al. Chromosome-scale, haplotype-resolved assembly of human genomes. *Nature biotechnology*, 39(3):309–312, 2021.

[32] E. Garrison and G. Marth. Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*, 2012.

[33] C. Gawad, W. Koh, and S. R. Quake. Single-cell genome sequencing: current state of the science. *Nature Reviews Genetics*, 17(3):175, 2016.

[34] C. Gawad, W. Koh, and S. R. Quake. Single-cell genome sequencing: current state of the science. *Nature Reviews Genetics*, 17(3):175, 2016.

[35] M. Gerlinger, A. J. Rowan, S. Horswell, J. Larkin, D. Endesfelder, E. Gronroos, P. Martinez, N. Matthews, A. Stewart, P. Tarpey, et al. Intratumor heterogeneity and branched evolution revealed by multiregion sequencing. *N Engl j Med*, 366: 883–892, 2012.

[36] M. Gerlinger, S. Horswell, J. Larkin, A. J. Rowan, M. P. Salm, I. Varela, R. Fisher, N. McGranahan, N. Matthews, C. R. Santos, et al. Genomic architecture and evolution of clear cell renal cell carcinomas defined by multiregion sequencing. *Nature genetics*, 46(3):225–233, 2014.

[37] M. Ghareghani, D. Porubskỳ, A. D. Sanders, S. Meiers, E. E. Eichler, J. O. Korbel, and T. Marschall. Strand-seq enables reliable separation of long reads by chromosome via expectation maximization. *Bioinformatics*, 34(13):i115–i123, 2018.

[38] T. Girardi, C. Vicente, J. Cools, and K. De Keersmaecker. The genetics and molecular biology of t-all. *Blood, The Journal of the American Society of Hematology*, 129(9):1113–1123, 2017.

[39] G. Glusman, H. C. Cox, and J. C. Roach. Whole-genome haplotyping approaches and genomic medicine. *Genome medicine*, 6(9):1–16, 2014.

[40] S. González-Barrera, F. Cortés-Ledesma, R. E. Wellinger, and A. Aguilera. Equal sister chromatid exchange is a major mechanism of double-strand break repair in yeast. *Molecular cell*, 11(6):1661–1671, 2003.

[41] S. Goodwin, J. Gurtowski, S. Ethe-Sayers, P. Deshpande, M. C. Schatz, and W. R. McCombie. Oxford nanopore sequencing, hybrid error correction, and de novo assembly of a eukaryotic genome. *Genome research*, 25(11):1750–1756, 2015.

[42] D. Gordon, J. Huddleston, M. J. P. Chaisson, C. M. Hill, Z. N. Kronenberg, K. M. Munson, M. Malig, A. Raja, I. Fiddes, L. W. Hillier, C. Dunn, C. Baker, J. Armstrong, M. Diekhans, B. Paten, J. Shendure, R. K. Wilson, D. Haussler, C.-S. Chin, and E. E. Eichler. Long-read sequence assembly of the gorilla genome. *Science*, 352(6281):aae0344, Apr. 2016.

[43] M. Hills, K. O'Neill, E. Falconer, R. Brinkman, and P. M. Lansdorp. Bait: organizing genomes and mapping rearrangements in single cells. *Genome medicine*, 5(9):82, 2013.

[44] M. Hills, E. Falconer, K. O'Neil, A. Sanders, K. Howe, V. Guryev, and P. M. Lansdorp. Construction Of Whole Genomes From Scaffolds Using Single Cell Strand-Seq Data. *bioRxiv*, page 271510, Feb. 2018.

[45] L. M. Hoffman, M. DeWire, S. Ryall, P. Buczkowicz, J. Leach, L. Miles, A. Ramani, M. Brudno, S. S. Kumar, R. Drissi, et al. Spatial genomic heterogeneity in diffuse intrinsic pontine and midline high-grade glioma: implications for diagnostic biopsy and targeted therapeutics. *Acta neuropathologica communications*, 4(1):1–8, 2016.

[46] W. Huber, J. Toedling, and L. M. Steinmetz. Transcript mapping with high-density oligonucleotide tiling arrays. *Bioinformatics*, 22(16):1963–1970, 2006.

[47] A. Janssen, M. van der Burg, K. Szuhai, G. J. Kops, and R. H. Medema. Chromosome segregation errors as a cause of dna damage and structural chromosome aberrations. *Science*, 333(6051):1895–1898, 2011.

[48] A. Janssen, M. van der Burg, K. Szuhai, G. J. Kops, and R. H. Medema. Chromosome segregation errors as a cause of dna damage and structural chromosome aberrations. *Science*, 333(6051):1895–1898, 2011.

[49] W. P. Kloosterman, M. Hoogstraat, O. Paling, M. Tavakoli-Yaraki, I. Renkens, J. S. Vermaat, M. J. van Roosmalen, S. van Lieshout, I. J. Nijman, W. Roessingh,

et al. Chromothripsis is a common mechanism driving genomic rearrangements in primary and metastatic colorectal cancer. *Genome biology*, 12(10):1–11, 2011.

[50] J. O. Korbel, A. E. Urban, J. P. Affourtit, B. Godwin, F. Grubert, J. F. Simons, P. M. Kim, D. Palejev, N. J. Carriero, L. Du, et al. Paired-end mapping reveals extensive structural variation in the human genome. *Science*, 318(5849):420–426, 2007.

[51] J. O. Korbel, A. E. Urban, J. P. Affourtit, B. Godwin, F. Grubert, J. F. Simons, P. M. Kim, D. Palejev, N. J. Carriero, L. Du, et al. Paired-end mapping reveals extensive structural variation in the human genome. *Science*, 318(5849):420–426, 2007.

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

[53] S. Koren, A. Rhie, B. P. Walenz, A. T. Dilthey, D. M. Bickhart, S. B. Kingan, S. Hiendleder, J. L. Williams, T. P. Smith, and A. M. Phillippy. De novo assembly of haplotype-resolved genomes with trio binning. *Nature biotechnology*, 36(12): 1174–1182, 2018.

[54] H. Lee and J.-S. Kim. Unexpected crispr on-target effects. *Nature biotechnology*, 36(8):703–704, 2018.

[55] M. L. Leibowitz, C.-Z. Zhang, and D. Pellman. Chromothripsis: a new mechanism for rapid karyotype evolution. *Annual review of genetics*, 49:183–211, 2015.

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

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

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

[59] H. Li and R. Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009.

[60] Y. Li, C. Schwab, S. L. Ryan, E. Papaemmanuil, H. M. Robinson, P. Jacobs, A. V. Moorman, S. Dyer, J. Borrow, M. Griffiths, et al. Constitutional and somatic rearrangement of chromosome 21 in acute lymphoblastic leukaemia. *Nature*, 508 (7494):98–102, 2014.

[61] Y. Li, N. D. Roberts, J. Weischenfeldt, J. A. Wala, O. Shapira, S. E. Schumacher, E. Khurana, J. Korbel, M. Imielinski, R. Beroukhim, et al. Patterns of structural variation in human cancer. *BioRxiv*, page 181339, 2017.

[62] Y. Lin, J. Yuan, M. Kolmogorov, M. W. Shen, M. Chaisson, and P. A. Pevzner. Assembly of long error-prone reads using de bruijn graphs. *Proceedings of the National Academy of Sciences*, 113(52):E8396–E8405, 2016.

[63] S. Ling, Z. Hu, Z. Yang, F. Yang, Y. Li, P. Lin, K. Chen, L. Dong, L. Cao, Y. Tao, et al. Extremely high genetic diversity in a single tumor points to prevalence of non-darwinian cell evolution. *Proceedings of the National Academy of Sciences*, 112(47):E6496–E6505, 2015.

[64] M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):550, 2014.

[65] J. Maciejowski, Y. Li, N. Bosco, P. J. Campbell, and T. de Lange. Chromothripsis and kataegis induced by telomere crisis. *Cell*, 163(7):1641–1654, 2015.

[66] B. R. Mardin, A. P. Drainas, S. M. Waszak, J. Weischenfeldt, M. Isokane, A. M. Stütz, B. Raeder, T. Efthymiopoulos, C. Buccitelli, M. Segura-Wang, et al. A cell-based model system links chromothripsis with hyperploidy. *Molecular systems biology*, 11(9):828, 2015.

[67] B. R. Mardin, A. P. Drainas, S. M. Waszak, J. Weischenfeldt, M. Isokane, A. M. Stütz, B. Raeder, T. Efthymiopoulos, C. Buccitelli, M. Segura-Wang, et al. A cell-based model system links chromothripsis with hyperploidy. *Molecular systems biology*, 11(9):828, 2015.

[68] B. McClintock. The production of homozygous deficient tissues with mutant characteristics by means of the aberrant mitotic behavior of ring-shaped chromosomes. *Genetics*, 23(4):315, 1938.

[69] B. McClintock. The stability of broken ends of chromosomes in zea mays. *Genetics*, 26(2):234, 1941.

[70] A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, et al. The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *Genome research*, 20(9):1297–1303, 2010.

[71] G. Mendel. Experiments in plant hybridization (1865). *Verhandlungen des naturforschenden Vereins Brünn.) Available online: www. mendelweb. org/Mendel. html (accessed on 1 January 2013)*, 1996.

[72] F. Mertens, B. Johansson, T. Fioretos, and F. Mitelman. The emerging complexity of gene fusions in cancer. *Nature Reviews Cancer*, 15(6):371–381, 2015.

[73] E. W. Myers. The fragment assembly string graph. *Bioinformatics*, 21(suppl_2): ii79–ii85, 2005.

[74] E. W. Myers, G. G. Sutton, A. L. Delcher, I. M. Dew, D. P. Fasulo, M. J. Flanigan, S. A. Kravitz, C. M. Mobarry, K. H. Reinert, K. A. Remington, et al. A whole-genome assembly of drosophila. *Science*, 287(5461):2196–2204, 2000.

[75] G. Myers. Efficient local alignment discovery amongst noisy long reads. In *International Workshop on Algorithms in Bioinformatics*, pages 52–67. Springer, 2014.

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

[77] N. E. Navin. Cancer genomics: one cell at a time. *Genome biology*, 15(8):452, 2014.

[78] S. Nik-Zainal, P. Van Loo, D. C. Wedge, L. B. Alexandrov, C. D. Greenman, K. W. Lau, K. Raine, D. Jones, J. Marshall, M. Ramakrishna, et al. The life history of 21 breast cancers. *Cell*, 149(5):994–1007, 2012.

[79] S. Nurk, B. P. Walenz, A. Rhie, M. R. Vollger, G. A. Logsdon, R. Grothe, K. H. Miga, E. E. Eichler, A. M. Phillippy, and S. Koren. Hicanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. *BioRxiv*, 2020.

[80] S. Nurk, S. Koren, A. Rhie, M. Rautiainen, A. V. Bzikadze, A. Mikheenko, M. R. Vollger, N. Altemose, L. Uralsky, A. Gershman, et al. The complete sequence of a human genome. *Science*, 376(6588):44–53, 2022.

[81] K. O'Neill, M. Hills, M. Gottlieb, M. Borkowski, A. Karsan, and P. M. Lansdorp. Assembling draft genomes using contiBAIT. *Bioinformatics*, May 2017.

[82] T. Onodera, K. Sadakane, and T. Shibuya. Detecting superbubbles in assembly graphs. In *International Workshop on Algorithms in Bioinformatics*, pages 338–348. Springer, 2013.

[83] A.-M. Patch, E. L. Christie, D. Etemadmoghadam, D. W. Garsed, J. George, S. Fereday, K. Nones, P. Cowin, K. Alsop, P. J. Bailey, et al. Whole–genome characterization of chemoresistant ovarian cancer. *Nature*, 521(7553):489–494, 2015.

[84] M. Patterson, T. Marschall, N. Pisanti, L. Van Iersel, L. Stougie, G. W. Klau, and A. Schönhuth. Whatshap: weighted haplotype assembly for future-generation sequencing reads. *Journal of Computational Biology*, 22(6):498–509, 2015.

[85] P. A. Pevzner, H. Tang, and M. S. Waterman. An eulerian path approach to dna fragment assembly. *Proceedings of the national academy of sciences*, 98(17): 9748–9753, 2001.

[86] D. Porubskỳ, A. D. Sanders, N. Van Wietmarschen, E. Falconer, M. Hills, D. C. Spierings, M. R. Bevova, V. Guryev, and P. M. Lansdorp. Direct chromosome-length haplotyping by single-cell sequencing. *Genome research*, 26(11):1565–1574, 2016.

[87] D. Porubskỳ, S. Garg, A. D. Sanders, J. O. Korbel, V. Guryev, P. M. Lansdorp, and T. Marschall. Dense and accurate whole-chromosome haplotyping of individual genomes. *bioRxiv*, page 126136, 2017.

[88] D. Porubsky, P. Ebert, P. A. Audano, M. R. Vollger, W. T. Harvey, P. Marijon, J. Ebler, K. M. Munson, M. Sorensen, A. Sulovari, et al. Fully phased human genome assembly without parental data using single-cell strand sequencing and long reads. *Nature biotechnology*, 39(3):302–308, 2021.

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

[90] A. Riches, C. Peddie, S. Rendell, P. Bryant, H. Zitzelsberger, J. Bruch, J. Smida, L. Hieber, and M. Bauchinger. Neoplastic transformation and cytogenetic changes after gamma irradiation of human epithelial cells expressing telomerase. *Radiation research*, 155(1):222–229, 2001.

[91] A. Riches, C. Peddie, S. Rendell, P. Bryant, H. Zitzelsberger, J. Bruch, J. Smida, L. Hieber, and M. Bauchinger. Neoplastic transformation and cytogenetic changes after gamma irradiation of human epithelial cells expressing telomerase. *Radiation research*, 155(1):222–229, 2001.

[92] A. Rode, K. K. Maass, K. V. Willmund, P. Lichter, and A. Ernst. Chromothripsis in cancer cells: An update. *International journal of cancer*, 138(10):2322–2333, 2016.

[93] M. Ronaghi, S. Karamohamed, B. Pettersson, M. Uhlén, and P. Nyrén. Real-time dna sequencing using detection of pyrophosphate release. *Analytical biochemistry*, 242(1):84–89, 1996.

[94] A. D. Sanders, M. Hills, D. Porubskỳ, V. Guryev, E. Falconer, and P. M. Lansdorp. Characterizing polymorphic inversions in human genomes by single-cell sequencing. *Genome research*, 26(11):1575–1587, 2016.

[95] A. D. Sanders, S. Meiers, M. Ghareghani, D. Porubskỳ, H. Jeong, M. A. C. van Vliet, T. Rausch, P. Richter-Pechańska, J. B. Kunz, S. Jenni, et al. Single-cell analysis of structural variations and complex rearrangements with tri-channel processing. *Nature Biotechnology*, 38(3):343–354, 2020.

[96] F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, J. C. Fiddes, C. Hutchison, P. M. Slocombe, and M. Smith. Nucleotide sequence of bacteriophage φx174 dna. *nature*, 265(5596):687–695, 1977.

[97] F. Sanger, S. Nicklen, and A. R. Coulson. Dna sequencing with chain-terminating inhibitors. *Proceedings of the national academy of sciences*, 74(12):5463–5467, 1977.

[98] S. Selvarajah, M. Yoshimoto, P. C. Park, G. Maire, J. Paderova, J. Bayani, G. Lim, K. Al-Romaih, J. A. Squire, and M. Zielenska. The breakage–fusion–bridge (bfb) cycle as a mechanism for generating genetic heterogeneity in osteosarcoma. *Chromosoma*, 115(6):459–467, 2006.

[99] S. P. Shah, A. Roth, R. Goya, A. Oloumi, G. Ha, Y. Zhao, G. Turashvili, J. Ding, K. Tse, G. Haffari, et al. The clonal and mutational evolution spectrum of primary triple-negative breast cancers. *Nature*, 486(7403):395–399, 2012.

[100] J. T. Simpson and R. Durbin. Efficient de novo assembly of large genomes using compressed data structures. *Genome research*, 22(3):549–556, 2012.

[101] A. Sottoriva, I. Spiteri, S. G. Piccirillo, A. Touloumis, V. P. Collins, J. C. Marioni, C. Curtis, C. Watts, and S. Tavaré. Intratumor heterogeneity in human glioblastoma reflects cancer evolutionary dynamics. *Proceedings of the National Academy of Sciences*, 110(10):4009–4014, 2013.

[102] M. Spielmann, D. G. Lupiáñez, and S. Mundlos. Structural variation in the 3d genome. *Nature Reviews Genetics*, 19(7):453–467, 2018.

[103] P. J. Stephens, C. D. Greenman, B. Fu, F. Yang, G. R. Bignell, L. J. Mudie, E. D. Pleasance, K. W. Lau, D. Beare, L. A. Stebbings, et al. Massive genomic rearrangement acquired in a single catastrophic event during cancer development. *cell*, 144(1):27–40, 2011.

[104] M. R. Stratton. Exploring the genomes of cancer cells: progress and promise. *science*, 331(6024):1553–1558, 2011.

[105] R. Tewhey, V. Bansal, A. Torkamani, E. J. Topol, and N. J. Schork. The importance of phase information for human genomics. *Nature Reviews Genetics*, 12 (3):215–223, 2011.

[106] S. L. Thompson, S. F. Bakhoum, and D. A. Compton. Mechanisms of chromosomal instability. *Current biology*, 20(6):R285–R295, 2010.

[107] T. J. Treangen and S. L. Salzberg. Repetitive dna and next-generation sequencing: computational challenges and solutions. *Nature Reviews Genetics*, 13(1): 36–46, 2012.

[108] T. Voet, E. Vanneste, N. Van der Aa, C. Melotte, S. Jackmaert, T. Vandendael, M. Declercq, S. Debrock, J.-P. Fryns, Y. Moreau, et al. Breakage–fusion–bridge cycles leading to inv dup del occur in human cleavage stage embryos. *Human mutation*, 32(7):783–793, 2011.

[109] Y. Wang, J. Waters, M. L. Leung, A. Unruh, W. Roh, X. Shi, K. Chen, P. Scheet, S. Vattathil, H. Liang, et al. Clonal evolution in breast cancer revealed by single nucleus genome sequencing. *Nature*, 512(7513):155–160, 2014.

[110] Y. K. Wang, A. Bashashati, M. S. Anglesio, D. R. Cochrane, D. S. Grewal, G. Ha, A. McPherson, H. M. Horlings, J. Senz, L. M. Prentice, et al. Genomic consequences of aberrant dna repair mechanisms stratify ovarian cancer histotypes. *Nature genetics*, 49(6):856–865, 2017.

[111] J. D. Watson and F. H. Crick. 1953. a structure for deoxyribose nucleic acid. *Nature*, 171:964–967, 1953.

[112] J. Weischenfeldt, O. Symmons, F. Spitz, and J. O. Korbel. Phenotypic impact of genomic structural variation: insights from and for human disease. *Nature Reviews Genetics*, 14(2):125–138, 2013.

[113] N. I. Weisenfeld, V. Kumar, P. Shah, D. M. Church, and D. B. Jaffe. Direct determination of diploid genome sequences. *Genome research*, 27(5):757–767, 2017.

[114] A. M. Wenger, P. Peluso, W. J. Rowell, P.-C. Chang, R. J. Hall, G. T. Concepcion, J. Ebler, A. Fungtammasan, A. Kolesnikov, N. D. Olson, et al. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nature biotechnology*, 37(10):1155–1162, 2019.

[115] M. J. Williams, B. Werner, C. P. Barnes, T. A. Graham, and A. Sottoriva. Identification of neutral tumor evolution across cancer types. *Nature genetics*, 48 (3):238–244, 2016.

[116] C. J. Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.

[117] L. R. Yates, M. Gerstung, S. Knappskog, C. Desmedt, G. Gundem, P. Van Loo, T. Aas, L. B. Alexandrov, D. Larsimont, H. Davies, et al. Subclonal diversification of primary breast cancer revealed by multiregion sequencing. *Nature medicine*, 21(7):751–759, 2015.

[118] M. Yoshihara, Y. Hayashizaki, and Y. Murakawa. Genomic instability of ipscs: challenges towards their clinical applications. *Stem Cell Reviews and Reports*, 13 (1):7–16, 2017.

[119] H. Zahn, A. Steif, E. Laks, P. Eirew, M. VanInsberghe, S. P. Shah, S. Aparicio, and C. L. Hansen. Scalable whole-genome single-cell library preparation without preamplification. *Nature methods*, 14(2):167, 2017.

[120] H. Zahn, A. Steif, E. Laks, P. Eirew, M. VanInsberghe, S. P. Shah, S. Aparicio, and C. L. Hansen. Scalable whole-genome single-cell library preparation without preamplification. *Nature methods*, 14(2):167, 2017.

[121] D. R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome research*, 18(5):821–829, 2008.

[122] C.-Z. Zhang, M. L. Leibowitz, and D. Pellman. Chromothripsis and beyond: rapid genome evolution from complex chromosomal rearrangements. *Genes & development*, 27(23):2513–2530, 2013.

[123] C.-Z. Zhang, A. Spektor, H. Cornils, J. M. Francis, E. K. Jackson, S. Liu, M. Meyerson, and D. Pellman. Chromothripsis from dna damage in micronuclei. *Nature*, 522(7555):179–184, 2015.

[124] J. Zhang, K. Kobert, T. Flouri, and A. Stamatakis. Pear: a fast and accurate illumina paired-end read merger. *Bioinformatics*, 30(5):614–620, 2014.