Title
Genome Rearrangements in Cancer and Human Genomes /

Permalink
https://escholarship.org/uc/item/921238hz

Author
Patel, Anand

Publication Date
2014

Peer reviewed|Thesis/dissertation
UNIVERSITY OF CALIFORNIA, SAN DIEGO

Genome Rearrangements in Cancer and Human Genomes

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Bioinformatics & Systems Biology

by

Anand Patel

Committee in charge:

Professor Vineet Bafna, Chair
Professor Yu-Tsueng Liu, Co-Chair
Professor Terry Gaasterland
Professor Pavel Pevzner
Professor Richard Schwab
Professor Christopher Woelk

2014
The dissertation of Anand Patel is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Co-Chair

Chair

University of California, San Diego

2014
DEDICATION

To my family. My parents set up a garden for me to grow in and have been extremely understanding of whatever eccentric personality I’ve adopted on the way to conduct research and complete this thesis.

Truly, this is the fruit of their support.

I’d like to dedicate this thesis to my grandfather, Surendra K. Patel. Ironically, he may be the biggest influential deterrent for completing graduate school. He was the only one in the family not to ask when I will graduate, but how I was enjoying life. Needless to say he will always be a role model in my life, for his humility, effective work ethic, genuine concern for others, and most of all will power. Few men have the intrinsic quality to be unjaded by troubles they face and find joy in every situation.
EPIGRAPH

To progress, and those who help the path.
TABLE OF CONTENTS

Signature Page ........................................ iii
Dedication ........................................... iv
Epigraph ............................................... v
Table of Contents .................................... vi
List of Figures ....................................... ix
List of Tables ....................................... xi
Acknowledgements ................................... xii
Vita .................................................... xiv
Abstract of the Dissertation ......................... xv

Chapter 1  Introduction ............................... 1

Chapter 2  Amplification and thrifty single molecule sequencing of recurrent somatic structural variations .... 9
2.1  Abstract ........................................ 9
2.2  Introduction .................................. 10
2.3  Results ....................................... 15
   2.3.1  Designing primers ......................... 15
   2.3.2  Sequencing amplified sequences harboring SVs . 18
   2.3.3  Identifying CDKN2A deletion given DNA break clustering ..................... 20
   2.3.4  Characterizing CDKN2A deletion assuming no DNA break clustering .......... 23
   2.3.5  Characterizing RUNX1-RUNX1T1 translocations .................. 24
   2.3.6  Dealing with tumor heterogeneity .......... 25
2.4  Discussion ................................... 26
2.5  Methods ..................................... 28
   2.5.1  AmBre: Primer generation and filtering 28
   2.5.2  AmBre: Primer selection with simulated annealing 29
   2.5.3  AmBre-analyze: PacBio sequence analysis ....... 30
   2.5.4  Experimental Methods .................... 34
2.6  Data Access .................................. 36
2.7  Tables ....................................... 37
2.8  Figures ...................................... 38
<table>
<thead>
<tr>
<th>Figure 2.1:</th>
<th>PAMP tiling design for capture of CDKN2A deletions</th>
<th>38</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.2:</td>
<td>AmBre pipeline with primer designing and PacBio long fragment sequence analysis.</td>
<td>39</td>
</tr>
<tr>
<td>Figure 2.3:</td>
<td>Designing of AMBRE-68</td>
<td>39</td>
</tr>
<tr>
<td>Figure 2.4:</td>
<td>Pacific Biosciences sequence analysis of breakpoints</td>
<td>40</td>
</tr>
<tr>
<td>Figure 2.5:</td>
<td>Gel electrophoresis of amplified CDKN2A deletion fragments in six cell lines</td>
<td>41</td>
</tr>
<tr>
<td>Figure 2.6:</td>
<td>Characterizing RUNX1-RUNX1T1 balanced translocation in Kasumi-1</td>
<td>42</td>
</tr>
<tr>
<td>Figure 2.7:</td>
<td>Filtering of local alignments from Pacific Biosciences sequencing</td>
<td>43</td>
</tr>
<tr>
<td>Figure 2.8:</td>
<td>Simulated annealing convergence for AMBRE-68 primer design</td>
<td>46</td>
</tr>
<tr>
<td>Figure 2.9:</td>
<td>Breakpoint estimates for A549, CEM, MCF7, and T98G from CGP. Last two rows are AMBRE-16 and AMBRE-68 input target regions.</td>
<td>46</td>
</tr>
<tr>
<td>Figure 2.10:</td>
<td>Gel electrophoresis for amplified CDKN2A deletions using AMBRE-16 design</td>
<td>47</td>
</tr>
<tr>
<td>Figure 2.11:</td>
<td>Breakpoint sequences for A549, CEM, Detroit562 and Sanger sequencing chromatogram of MCF7 and T98G</td>
<td>47</td>
</tr>
<tr>
<td>Figure 2.12:</td>
<td>Pacific Biosciences sequence coverage of CDKN2A deletion amplicons</td>
<td>48</td>
</tr>
<tr>
<td>Figure 2.13:</td>
<td>DNA flexibility around CDKN2A deletion breaks</td>
<td>51</td>
</tr>
<tr>
<td>Figure 2.14:</td>
<td>Spike-in heterogeneity experiment for capturing CDKN2A deletion breakpoints in A549 and MCF7 with unmutated DNA</td>
<td>53</td>
</tr>
<tr>
<td>Figure 2.15:</td>
<td>Spike-in heterogeneity experiment for capturing CDKN2A deletion breakpoints in A549 and Detroit562</td>
<td>54</td>
</tr>
<tr>
<td>Figure 3.1:</td>
<td>Hypothetical chromosome undergoing chromothripsis</td>
<td>77</td>
</tr>
<tr>
<td>Figure 3.2:</td>
<td>Following Stephens et al. simulation procedure, a sequence of possible rearrangements steps to explain the observed breakpoints in Table 3.1</td>
<td>78</td>
</tr>
<tr>
<td>Figure 3.3:</td>
<td>Simulations of progressive rearrangements by sampling observed breakpoints</td>
<td>79</td>
</tr>
<tr>
<td>Figure 3.4:</td>
<td>Charts of breakpoints versus copy number states for simulations with an overrepresentation of inversions.</td>
<td>80</td>
</tr>
<tr>
<td>Figure 3.5:</td>
<td>Simulation of progressive rearrangements from a simulated inversion and deletion only chromosome</td>
<td>80</td>
</tr>
<tr>
<td>Figure 3.6:</td>
<td>Observed breakpoints on chromosome 5 of cell line TK10 explained by progressive rearrangements</td>
<td>81</td>
</tr>
<tr>
<td>Figure 3.7:</td>
<td>Chromosome rearranged by only inversions and deletions</td>
<td>83</td>
</tr>
<tr>
<td>Figure 3.8:</td>
<td>Transposition of ‘de’ into ‘gh’</td>
<td>85</td>
</tr>
</tbody>
</table>
Figure 3.9: Complex genome rearrangement with three breakpoints. Poly-fusion event in chromosome 8 involving three genes discovered by McPherson et al. (2012). ... 86
Figure 3.10: Complex genome rearrangement event across multiple chromosomes as discovered by McPherson et al. (2012) ... 88
Figure 3.11: Multiple scenarios of progressive rearrangements explain rearranged chromosomes ... 90

Figure 4.1: Variant phasing problem in diploid genomes ... 118
Figure 4.2: Comparison of proximity-ligation read data with simulated read data ... 119
Figure 4.3: Comparison of proximity-ligation read data with simulated read data in 40kb windows ... 120
Figure 4.4: Deletion classification accuracy ... 121
Figure 4.5: Importance of proximity-ligation sequencing in deletion classification ... 121
Figure 4.6: Classification accuracy tolerance to deletion breakpoint error ... 122
Figure 4.7: Classification accuracy in deletions called by Mills et al. ... 122
Figure 4.8: Classifier scoring corresponds with classification accuracy ... 123
Figure 4.9: SKAβ gene with suspicious deletion predictions ... 124
Figure 4.10: Assigning deletion phasings using pedigree ... 132
Figure 4.11: Classification accuracy when increasing simulated read coverage ... 133
Figure 4.12: Example of HapMap3 homozygous deletion inconsistent with SVM call ... 134
Figure 4.13: Phasing to incorrect error in mills appears in SKAβ gene ... 134
Figure 4.14: Phasing to hom error in real ... 135
Figure 4.15: Difficult to distinguish error ... 135
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>CDKN2A deletion breakpoints in cancer cell lines</td>
<td>37</td>
</tr>
<tr>
<td>3.1</td>
<td>Breakpoints of a rearranged chromosome in Figure 3.1.</td>
<td>75</td>
</tr>
<tr>
<td>3.2</td>
<td>Fraction of chromosomes in Figure 3.3a with few copy number states for given breakpoint counts.</td>
<td>75</td>
</tr>
<tr>
<td>3.3</td>
<td>Counts of breakpoints explained by progressive rearrangements in chromothripsis chromosomes</td>
<td>76</td>
</tr>
<tr>
<td>4.1</td>
<td>Proximity ligation and whole genome read coverage for improving phasing of deletions</td>
<td>116</td>
</tr>
<tr>
<td>4.2</td>
<td>Confusion tables for RandomForest and SVM on Mills et al. deletion release set</td>
<td>117</td>
</tr>
<tr>
<td>4.3</td>
<td>Mapped read counts from simulated chromosomes.</td>
<td>129</td>
</tr>
<tr>
<td>4.4</td>
<td>HapMap3 data is insufficient for training models</td>
<td>130</td>
</tr>
<tr>
<td>4.5</td>
<td>Inconsistencies between SVM calls and HapMap3 deletion labels</td>
<td>131</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

No research effort is the work of a single individual. This thesis is no different. I would like to thank my mentor, Dr. Vineet Bafna. I had the opportunity to be groomed for bioinformatics as an undergraduate by Dr. Bafna and am most grateful for Dr. Bafna’s encouragement to work on diverse research areas and letting me make mistakes in each project. Dr. Bafna holds his students to a high caliber of scientific research, and often I forget this caliber is necessary for ethical research and respect as a scientist. In addition to teaching problem solving and computational techniques expected of a bioinformatics data scientist, Dr. Bafna demonstrated how passionate scientists should behave in scientific arguments and meetings. In these situations, a calm and delicate demeanor is important for quickly arriving at scientific truths, which in the end is the true goal of any passionate scientist.

I’d also like to acknowledge the advising I have received from Dr. Liu. Dr. Liu was instrumental in my molecular biology research. I acquired confidence through his trust in my ability to design and carryout experiments in the wet laboratory. I’m also thankful for the long hours spent with Dr. Liu interpreting results and speculating on invisible molecular events that could explain the results. I’d also like to acknowledge the rest of my thesis committee for their guidance and encouragement on a variety of educating research projects and life choices; Dr. Richard Schwab, Dr. Terry Gaasterland, Dr. Christopher Woelk, and Dr. Pavel Pevzner.

At UCSD, I’ve had the pleasure of working along side numerous bright scientists whose expertise ranged from specific programming languages, genomics, proteomics, and biochemical techniques. My tenure at UCSD has been long and the list of scientists a party to both scientific and non-scientific helpful discussions is
equally long. The shortest acknowledgement list I can live with is: Dr. Sam Payne, Dr. Natalie Castellana, Dr. Nitin Gupta, Dr. Son K. Pham, Stefano Bonissone, Dr. Nitin Udpa, Dr. Paul Medvedev, Dr. Shay Zakov, Dr. Harish Nagarajan, Dr. Marcus Kinsella, Christine Lo, Dr. Joceylne Bruand, Roy Ronen, Eric Scott, Siddarth Selvaraj, and Viraj Deshpande. My experience at UCSD would have felt void without any of these fellow scientists.

Chapter 2 is adapted from Patel A, Schwab R, Liu Y, Bafna V. Amplification and thrifty single-molecule sequencing of recurrent somatic structural variations. Genome Research. (2013). The dissertation author was the primary author of this paper, and was responsible for the research.

Chapter 3 is adapted from Kinsella M, Patel A, Bafna V. The elusive evidence for chromothripsis. Nucleic Acids Research. (2014). The dissertation author was a joint first author of this paper, and was responsible for computational research with regards to breakpoint analysis.

Chapter 4 is from material in preparation for publication in Bioinformatics, Patel A, Bafna V. InPhaDel: Integrative whole genome and proximity-ligation sequencing. The dissertation author was the primary author of this paper, and was responsible for the research.
VITA

2009 B.S. in Computer Science, University of California, San Diego

2014 Ph. D. in Bioinformatics & Systems Biology, University of California, San Diego

PUBLICATIONS


ABSTRACT OF THE DISSERTATION

Genome Rearrangements in Cancer and Human Genomes

by

Anand Patel

Doctor of Philosophy in Bioinformatics & Systems Biology

University of California, San Diego, 2014

Professor Vineet Bafna, Chair
Professor Yu-Tsueng Liu, Co-Chair

In recent decades, advances in sequencing technologies have led to an explosion of discoveries in cancer. While observing large chromosomal abnormalities under the microscope has demonstrated genome rearrangements can drive cancer progression, more recent technologies enabled discoveries of mutations private to single cancer patients and uncovered a broader mutation diversity. My dissertation introduces novel connections between computational methods and sequencing techniques to solve open problems in genome rearrangement research.

To improve non-invasive cancer monitoring, genome rearrangements can
serve as the ideal cancer biomarker for accurately monitoring tumor burden and catching relapse earlier. My approach, AmBre (Amplification of Breakpoints), characterizes a target genome rearrangement’s breakpoints for use as a quantitative marker in measuring amounts of tumor DNA. For a target genome rearrangement such as \textit{CDKN2A} deletion, AmBre accounts for diverse deletion breakpoints and amplifies any DNA harboring the \textit{CDKN2A} deletion. Since only the tumor DNA is amplified, breakpoints can be detected in tissues or blood with little tumor DNA in high background of unmutated DNA. Furthermore, AmBre relies on sequencing technologies to read the enriched DNA. For parallel detection of breakpoints across numerous samples, a geometry based rearrangement caller was developed to handle long reads generated by Pacific Biosciences sequencing instruments.

In addition, I will discuss the limitations of sequencing technologies in inferring mechanisms for rearranging genomes. Specifically, sequencing data alone cannot infer a complex cancer chromosome was formed by a single shattering and repair mechanism (chromothripsis) or a series of progressive rearrangements.

Lastly, genomes are diploid and genome rearrangements can appear on one or both homologous chromosomes. Detecting genome rearrangements is challenging and inferring which chromosome is affected by the rearrangement is even more difficult. Having already called genome rearrangements such as deletions, I will show how proximity-ligation sequencing can be repurposed to assign deletions to a chromosome by phasing deletions with variants. In effect, my endeavors in genome rearrangement research show the field is constantly evolving with advances being made by complementing sequencing strategies and computational methods.
Chapter 1

Introduction

Remarkable progress has been made in understanding DNA. All living organisms store their inheritable genetic information in DNA and a human genome is organized in 44 autosomal chromosomes and 2 sex chromosomes. These DNA sequences have extraordinary similarity to other species, which is explained by Darwinian evolution. Ancient organisms reproduced and selected for beneficial traits. After many generations, diverse organisms are produced—all from a common origin. Hence, different organisms have different characteristics, but similar DNA. While a representative species genome can distinguish between humans and other primates, the genomes for any two humans are not identical. The variations found in the human genome can be responsible for inherited physical characteristics, personality attributes, and risks for disease.

Cancer is another example of an evolutionary process in the human genome, demonstrating exponential growth on the scale of a human lifespan. Cancer is a disease where cells uncontrollably proliferate. In humans, there are safety mechanisms to prevent cancerous behavior or destroy cancer cells. Cancer cells that are able to evade these safety mechanisms are naturally selected for and continue
to grow through cancer cell divisions. Most of the cancerous behaviors are acquired through DNA mutations, which are passed on to other cancer cells in subsequent cell divisions. For example, a gene that suppresses tumor growth by reducing proliferation may be inactivated by a DNA mutation. The DNA mutation increases fitness for the survival cancer cells and is passed on to any cancer progeny.

In cancer genome evolution, there are different kinds of variation present in DNA sequences. The easiest kind of variation to conceptualize is single nucleotide differences—that is, two DNA sequences comprised of As, Cs, Gs, and Ts differ at a single position on the sequence. It was shown recently, acquiring a single nucleotide mutation at a specific position of the KRAS gene modifies the KRAS protein’s function to continually recruit growth factors, regardless of external environment signals (i.e., uncontrolled proliferation). A different form of cancer variation is genome rearrangements. The most famous example is the BCR-ABL gene fusion, found in 95% of chronic myeloid leukemia patients. BCR, which lies on chromosome 22 translocates with ABL on chromosome 9. The translocation forms a large chromosome and a mutant small chromosome known as the Philadelphia chromosome, which is identifiable under a microscope.

Only recently have we been able to make the connection between specific genome variations and their functional impact to the cell. A feat only been possible with novel molecular biology technologies. A little over fifty years ago, the double-helix structure of DNA was resolved by Waston, Crick, Wilkins, and Franklin and today we have complete genome sequences for numerous people. “A serious assessment of the history of technology reveals that technological change is exponential. Exponential growth is a feature of any evolutionary process, of which technology is a primary example.” (Kurzweil, 2005, pg. 12). In a converse harmony, the exponential growth of technologies rapidly enhanced our understanding of
the genome evolution. This thesis will build upon sequencing technologies using computational methods to improve our understanding of rearrangements in human and cancer genomes.

The first technology used to detect large genome rearrangements, particularly in cancer, is karyotyping of condensed chromosomes. Originally, condensed chromosomes were stained with Giemsa for contrast and observed under a microscope. Giemsa stains AT-rich regions more darkly than GC-rich regions, creating unique banding patterns for each chromosome. The condensed chromosomes from a nucleus can be separated and analyzed for deletions, translocations, and other chromosomal abnormalities. The resolution for detecting chromosomal abnormalities has greatly improved with fluorescent in situ hybridization (FISH), but FISH can still only detect 1mb or greater rearrangements. In a new generation of technologies, high-throughput probe hybridization arrays have enabled better resolution, capable of detecting 10kb or greater deletions and duplications. The hybridization arrays are unable to detect inversions and translocations, and karyotyping is still necessary.

In the last ten years, developments in sequencing have vastly accelerated our understanding of cancer mutations. Sequencing has revolutionized detection of single nucleotide differences in cancer, previously unfeasible by probe hybridization methods. With sequencing, even small insertions and deletions of 1 to 5 nucleotides can be reliably discovered. To identify mutations and adequately sample the entire genome, sequencing technologies have taken a shotgun approach where millions of short DNA segments are read, and computationally reconstituted into full chromosomal sequences. Advances in sequencing have been able to extend the read length from 36bp (Illumina’s Genome Analyzer 2010) to 300bp (Illumina’s MiSeq 2014). This fine scale of sequencing has even facilitated discoveries of genome rearrangements, although with much higher false positive rates than single
nucleotide or small indel differences. The reason for the difference in accuracy is inherent to the shotgun approach. Single nucleotide and small indels can be found within single reads. Errors within single reads can be mistaken for mutations, however each genome locus is sampled multiple times and this type of error is mitigated. Another possible error is mutations gone unnoticed in repetitive regions. Short reads have trouble being accurately matched to highly repetitive sequences. However, technological developments have increased read lengths, which enables better read matching to repetitive sequences.

Detection of genome rearrangements is challenging for a number of reasons, including the ones faced by small nucleotide variations. There is a great diversity of kinds of genome rearrangements and has a correspondingly confusing terminology with extensive definitions. For example, within the genome rearrangement category falls deletions, duplications, tandem duplications, interspersed duplications, segmental duplications, interchromosomal translocations, intrachromosomal translocations, insertions, inversions, complex rearrangements, chromothripsis, and the list goes on infinitium. Sequencing technologies paired with complementing computational methods yield different accuracies in detecting different types of rearrangement. For example, hybridization arrays can only identify copy number variations, such as deletions and duplications. Also, the location of the copy number variation can be imprecise by 10kb as shown in Chapter 2. To further confuse the matter, some of these definitions allude to molecular breakage events that occurred in a cell or ancestor cell to create the observed cancer genome.

To simplify, recall each cell has two copies of each chromosome. Rearrangements are differences in a comparison of a chromosome or set of chromosomes to some reference set of chromosomes. This perspective is immensely helpful. Rooted in the 1980s, genome rearrangement theory has been long established in
the organism–to–organism comparative genomics community. For application to cancer and a single species genomes, there are caveats with this perspective that need to be considered. Sequencing technologies, including hybridization techniques, are methods trying to identify the diploid genome of cells. Cancer presents a unique difficulty as tumor tissue samples are a mixture of normal cells and a heterogenous population of cancer cells. Current sequencing technologies only offer hints at the diploid genomes of each of the cancer cells. The core of my thesis is to address some of these caveats with genome rearrangement identification in the age of high-throughput sequencing. In the following sections, I will discuss an assay for targeting rearrangements in cancer, cellular mechanisms that create rearrangements in cancer, and finally a method to obtain a complete diploid genome from sequencing.

Observing chromosomal abnormalities using FISH is current standard practice for diagnosis of numerous cancers and only recently has sequencing taken hold in diagnosis and influencing clinical decisions. In Chapter 2, we expand upon the idea of using chromosomal abnormalities as a biomarker for non-invasive cancer monitoring. The current standard of care for cancer patients post-surgery or therapy is MRI, PET, or CT-scan imaging and protein-based biomarkers, such as PSA and CA-153. The imaging technologies are expensive and low resolution, whereas the protein biomarkers are cheaper, but unreliable. Targeting tumor DNA mutations is more reliable and cost-effective with the potential to catch cancer relapse or remission earlier than the current standard of care. My solution is to target a cancer genome rearrangement and secondly use the patient’s unique rearranged sequence to monitor their tumor burden. My technology is based on amplification of genome rearrangement breakpoints (AmBre). Novel molecular biology and computational algorithms are applied to improve amplification of genome rearrangements com-
pared to PAMP. In addition, I developed computational methods to analyze genome rearrangements on the single-molecule sequencing Pacific Biosciences platform. The Pacific Biosciences instrument is low-throughput, costly, and has a high error rate, but offers worthwhile long read (10kb or more) capabilities. Long reads, even though with small errors, trivializes the task for accurately calling larger genome rearrangements.

Discovery commences with the awareness of anomaly, i.e., with the recognition that nature has somehow violated the paradigm-induced expectations that govern normal science.

Kuhn 1970

In sequencing numerous cancer cell-lines, a group at the Sanger Institute observed a chromosome profoundly impacted with rearrangements, yet the copy number appeared to oscillate between zero and one. They explained this anomaly by hypothesizing chromothripsis, a novel DNA mutating mechanism where the chromosome shatters and repairs itself. As the mechanism encourages mutations, any beneficial mutations to the cancer are selected for. In Chapter 3, I examine the statistical method used to describe chromothripsis as an anomaly. Additionally, I show sequencing data alone cannot distinguish between the chromothripsis hypothesis and rearrangements occurring in separate cell divisions. The progressive rearrangement scenario is the widely-accepted normal mechanism for cancer genome evolution.

Each human cell has pairs of nearly identical chromosomes; each chromosome set of 22 autosomal chromosomes and one sex chromosome is donated by each parent. Shotgun sequencing approaches complicate the process of linking mutations that appear on the same chromosome or homologous chromosomes. Knowing whether mutations are on the same chromosome or homologous chromosomes has been shown to be important in studies of the genetic basis for rare diseases
and neurological disorders like autism spectrum disorder and schizophrenia. In sequencing experiments of a single individual affected by a rare disease, thousands of variants are observed and falsely implicate numerous genes being involved with the rare disease under study. A more substantial evincer for implicating a gene is to observe rare variants appearing in the single gene but on both homologous chromosomes. Rare variants and greatly reduced gene function is a better fitting hypothesis to the cause of a rare disease. For each gene knowing whether mutations fall on the same or homologous chromosome can only be informed by phasing mutations to other variants on the same chromosome. In the final chapter 4, I examine how to phase variants to genome rearrangements, specifically deletions. Previously, sequencing of entire pedigrees was necessary to phase variants, since parents transmit variants as complete chromosomes to their child. In lieu of extensive pedigree sequencing, I rely on only whole genome sequencing and a recently developed proximity-ligation sequencing for the single individual under study. In an integrative approach to phase deletions (InPhaDel), artificial intelligence techniques were applied to both sources of sequencing data to learn models for predicting deletion phasings. To boost the performance of these models, I developed a proximity-ligation read simulator.

The components of my thesis clearly show the synchronization between technology and progress in research in human genome variation. Today, we use computational methods to infer diploid genomes from signals produced by sequencing instruments. As sequencing technologies improve, the computational methods as described in this thesis may become antiquated. We do not have to look far into the future to find such disruptive technologies. Single cell and single-molecule sequencing are already frontiering this effort. These transformative sequencing technologies will take advantage of special biochemical or physical properties of
DNA. Often, these instruments will create new computational problems. A lasting achievement in my thesis is AmBre. AmBre has the potential to overcome limitations in biological samples and revolutionize how we monitor for cancer relapse and delineate malignant tumor growths earlier.
Chapter 2

Amplification and thrifty single molecule sequencing of recurrent somatic structural variations

2.1 Abstract

Deletion of tumor suppressor genes as well as other genomic rearrangements pervade cancer genomes across numerous different types of solid tumor and hematologic malignancies. However, even for a specific rearrangement, the breakpoints may vary between individuals, such as the recurrent CDKN2A deletion. Characterizing the exact breakpoint for structural variants (SVs), has utility as patient specific tumor biomarkers. We propose AmBre (Amplification of Breakpoints), a method to target SV breakpoints occurring in samples composed of heterogeneous tumor and germline DNA. Additionally, AmBre validates SVs called by whole exome/genome sequencing and hybridization arrays. AmBre involves a PCR-based approach to amplify the DNA segment containing a SV’s breakpoint and then confirms breakpoints
using sequencing by Pacific Biosciences. To amplify breakpoints with PCR, primers tiling specified target regions are carefully selected with a simulated annealing algorithm to minimize off-target amplification and maximize efficiency at capturing all possible breakpoints within the target regions. To confirm correct amplification and obtain breakpoints, PCR amplicons are combined without barcoding and long-read sequenced simultaneously using a single molecule real-time sequencing (SMRT) cell. Our algorithm efficiently separates reads based on breakpoints. Each read group supporting the same breakpoint corresponds with an amplicon and a consensus amplicon sequence is called. AmBre was used to discover CDKN2A deletion breakpoints in cancer cell lines: A549, CEM, Detroit562, MOLT4, MCF7 and T98G. Also, we successfully assayed RUNX1-RUNX1T1 reciprocal translocations by finding both breakpoints in the Kasumi-1 cell line. AmBre successfully targets SVs where DNA harboring the breakpoints are present in 1:1000 mixtures.

2.2 Introduction

Cancer develops through a series of genetic mutations, with tumor cells acquiring pernicious mutations that eventually lead to metastatic disease. The DNA mutations contributing to oncogenesis are not limited to point mutations, but include large chromosomal rearrangements, duplications, and deletions. It has been suggested that recurring mutations are the likely drivers for cancer, and might be viable biomarkers for disease detection and prognosis. For instance, a translocation occurs between chromosome 21 and 8 that fuses RUNX1 and RUNX1T1 genes in 12% of acute myeloid leukemia (AML) cases (Xiao et al., 2001). The fusion results in a chimeric oncoprotein. The chimeric protein contributes to initial leukemia cell growth mostly through transcriptional repression of wild-type RUNX1
targets (Downing, 1999). Alternatively, the loss of DNA may also contribute to cancer progression. For example, many human cancers frequently delete chromosome 9p21-22 locus containing \textit{MTAP}, \textit{CDKN2A}, and \textit{CDKN2B} genes. The locus encodes INK4 proteins (p15\textsuperscript{INK4B}, p16\textsuperscript{INK4A}) that inhibit cyclin-dependent kinases, CDK4 and CDK6, and p14\textsuperscript{ARF}, which inactivates MDM2 and thereby regulating TP53. Thus, expression of these proteins is responsible for G1 cell-cycle arrest and independently signaling apoptosis (Wessely, 2010; Kim et al., 2012). Homozygous deletions frequent the 9p21-22 locus, in particular \textit{CDKN2A}, which encodes both p16\textsuperscript{INK4A} and p14\textsuperscript{ARF}, as the single event diminishes expression of multiple proteins each with unique tumor suppressor activity.

In a clinical setting, driver DNA lesions can be used to (a) detect tumor DNA in individuals and (b) monitor tumor burden during or after treatment. (Michor et al., 2005) and (Bartley et al., 2010) demonstrated how identification of \textit{BCR-ABL1} gene fusion at the DNA level in leukemia patients leads to a more sensitive test for measuring tumor burden than current \textit{BCR-ABL1} mRNA tests. Measuring changes in tumor burden during therapeutic treatment is critical for checking therapy effectiveness and deciding to continue treatment. Their approach focuses on the frequent translocation of \textit{BCR-ABL1} in leukemia and has not been applied to solid tumors. In a more recent study, circulatory biomarkers were assessed in their ability to monitor metastatic breast cancer (Dawson et al., 2013). The researchers applied a variety of sequencing methods to identify point mutations in \textit{PIK3CA} and \textit{TP53} and other somatic structural variations for use as circulatory tumor DNA markers. They found that circulatory tumor DNA had the highest correlation with tumor burden and greater dynamic range than current standard of care CA 15-3 biomarker and circulatory tumor cell counting.

These studies all focused on tumor burden monitoring \textit{after} the specific lesion
had been fully characterized. While monitoring is easy for point mutations and structural variants with known breakpoints, it is very difficult when the breakpoint of the structural variation is not known. At the same time, large variants are potentially much more specific for tumor detection and monitoring, and a test that could identify them reliably would have higher sensitivity for monitoring tumor burden. Reliable and sensitive identification of breakpoints in tumor DNA could also serve as a diagnostic for early detection.

Whole genome sequencing experiments (analyzed with appropriate tools like BreakDancer (Chen et al., 2009), Pindel (Ye et al., 2009), and SVDetect (Zeitouni et al., 2010)) have the potential to identify point mutations and structural variations in individual samples. However, clinical tumor samples are a mixture of tumor cells and normal cells and require ultra-deep sequencing to analyze tumor DNA.

Therefore, current approaches apply ultra-deep sequencing after targeted amplification of select genes (Harismendy et al., 2011). Unfortunately, these methods are unable to reliably identify structural variation with uncertain breakpoints. Alternatively, DNA hybridization microarrays (SNP-arrays), which are still widely used in clinics, are capable of calling copy number variation, from which deletions and gene amplifications can be inferred. However, the technology is only reliable with homogeneous samples and only reports low resolution boundaries estimates (Greenman et al., 2010), insufficient for performing tumor burden monitoring assays. Thus, a challenge remains how to detect DNA markers, specifically, somatic structural variations, in a complex patient sample containing a mixture of tumor DNA and germline DNA. This is particularly challenging when the exact breakpoints are needed for quantitative DNA assays.

To identify unknown DNA breakpoints associated with known translocations and deletions, we describe a pipeline *AmBre (Amplification of Breakpoints)*, which
builds upon the PAMP approach (Liu and Carson, 2007). PAMP is a PCR assay, developed to selectively amplify the tumor DNA sequence containing a structural variation. To illustrate how PAMP works, consider a deletion on chr9 (\textit{CDKN2A} locus) with unknown breakpoints located around the \textit{CDKN2A} gene. Illustrated in Figure 2.1, a tiling of evenly spaced forward (blue arrows) primers and reverse primers (red arrows) are selected around the \textit{CDKN2A} gene. The spacing between primers is approximately 1kb apart. The innermost forward and reverse primers are distantly spaced such that they will not amplify sequence from germline DNA.

All tiling primers are used in a single multiplex PCR. Any \textit{CDKN2A} deletion in the tumor DNA will lead to a forward and reverse primer being proximally located (< 2kb) on the tumor DNA, resulting in a targeted DNA amplification of the tumor DNA harboring the deletion, but not germline DNA. This strategy takes advantage of polymerases having a limited amplifying length and genomic rearrangements within tumor DNA resulting in novel adjacencies of germline DNA sequences for selective and sensitive amplification of tumor DNA over germline DNA.

Although it has potential, PAMP has challenges. In the multiplexed reaction, all primers must be evenly spaced so as to amplify any deletion in the region and primer pairs cannot dimerize. In a large (say, 100kb) region, this implies we need to find a design of 100 applicable primers from a large candidate set of over 5000 potential primers. An exhaustive search of all candidate primer combinations is infeasible (5000 candidate primers and 50 to 100 primers desired would result in searching $\sum_{50 \leq i \leq 100} \binom{5000}{i} \approx 10^{211}$ combinations). Bashir et al. formulated PAMP primer tiling as a computational problem and defined a cost associated with each subset of candidate primers (Bashir et al., 2007). Furthermore, the authors showed simulated annealing (Kirkpatrick, 1984) could efficiently find low cost PAMP primer
designs for contiguous breakpoint regions. Even with these improvements, PAMP is limited to recurrent structural variations where breakpoints appear in short breakpoint regions (< 40kb), as a large number of primers in a single reaction inevitably leads to loss of sensitivity with off-target DNA synthesis and increased spurious primer-primer interactions. Finally, PAMP detects the amplified product and identifies breakpoints via DNA hybridization arrays (Bashir et al., 2009) which had the additional challenge of designing probes that match the primer designs.

**Overview of AmBre and Results.** AmBre resolves these issues with a three phase approach (Figure 2.2). The first (AmBre-design) involves a revised computational approach to designing multiplex primers on discontiguous DNA regions ignoring regions known to not contain breakpoints. This requires some changes to the optimization function and results in a more flexible design with better performance on sparse regions. The output of this phase is a collection of primers that can be mixed in a single multiple primer reaction.

In the second, experiment phase (AmBre-amplify), long range PCR amplifies target amplicons, which reduces the number of primers required in a single reaction. For example, PAMP, using their proposed traditional PCR, would require 600 primers to cover a 600kb region, with over 180,000 putative interactions. In contrast, to cover the same region, AmBre would need < 100 primers with only 5,000 possible interactions, which improves reliable amplification from proposed designs. In AmBre, the amplified products are sequenced using the Pacific Biosciences real-time sequencing (PacBio RS) platform (English et al., 2012). Our analysis allows us to mix the amplicons prior to sequencing, with computational separation of breakpoints in the third phase.

The final, computational phase (AmBre-analyze) involves a customized
analysis of sequenced reads to identify DNA breakpoints for each tumor genome. The analysis involves clustering of split mapped reads followed by error correction, and sequence reconstruction around the breakpoint regions. We demonstrated that AmBre can successfully detect targeted structural variations (potential tumor DNA biomarkers) by identifying $CDKN2A$ deletion breakpoints in the cancer cell lines A549, CEM, Detroit562, MCF7, MOLT4, and T98G. AmBre resolved breakpoints for MCF7 and T98G, which had not been previously discovered by other studies. Furthermore, AmBre easily extends to identify translocations and inversions, which is demonstrated here with $RUNX1$-$RUNX1T1$ translocation in the cancer cell line Kasumi-1.

## 2.3 Results

### 2.3.1 Designing primers

The input to AmBre-design is a collection of genomic intervals for the forward region, denoted by $F$, a collection of genomic intervals for the reverse region ($R$), and parameter $d$. The output is a collection of forward primers in $F$ and reverse primers located in $R$ spaced apart by approximately $d$. AmBre-design has the following steps:

- Candidate primer generation from target breakpoint regions, where oligonucleotides are selected according to thermodynamic properties. Primers with significant self-dimerization are eliminated. Primer pairs that are likely to dimerize or cause off-target amplifications are marked as incompatible (Methods).

- The list of candidate primers and incompatible primer pairs are used to design
Denote a primer design, $P$ as a subset of candidate primers numbered according to the order of genomic start locations $l_1, l_2, l_3, \ldots l_n$. Let set $E$ denote incompatible primer pairs. We associate a cost $C(P)$ with each design, and seek to find designs with minimum cost. Our formulation of cost differs from Bashir et al. to accommodate sparser primer designs and targeting discontiguous regions (see 2.9.1). The parameter $d$ is set to be half the maximum feasible PCR amplicon size. Thus, for long-range polymerases used here, we use $d = 6500$, corresponding to a desirable amplicon size $\leq 13$kb. The cost of the design is a sum of incompatibility costs for each pair, and coverage costs. 

For the coverage, let $\Delta_i(P) = l_{i+1} - l_i$ denote the gap between adjacent pairs. If $\Delta_i(P) > d$, we run the risk of the product being too long to be amplified. On the other hand, if $\Delta_i(P) << d$, we have a design with extra primers that greatly decrease the efficiency of the reaction. Let parameter $\rho$, with $0 < \rho \leq 1$, describe a target density $1 + \rho$ of primers every $d$ bp, corresponding to a primer every $\frac{d}{1+\rho} \simeq (1 - \rho)d$ bp. Ideally, the distance between adjacent primers is bounded by $(1 - \rho)d \leq \Delta_i(P) \leq d$. A design is penalized if the distances violate these constraints. Formally,

\[
C(P) = \sum_{(i,j) \in E} w_p + \sum_i \max\{\Delta_i(P) - d, 0, (1 - \rho)d - \Delta_i(P)\} \quad (2.1)
\]

Experiments revealed that even a single incompatible pair severely diminishes the multiple primer reaction (Bashir et al., 2007). Therefore, we set $w_p = \infty$ for our designs. We empirically choose $\rho = 0.2$. Similar to Bashir et al., simulated annealing is used to find low cost primer designs by applying our cost function (Figure 2.3
The algorithm explores the large space of all primer designs by initiating a random primer subset and improving the primer subset with iterative addition or removals of primers. Since the algorithm involves randomization and has parameters governing convergence to low cost designs, simulated annealing is repeated multiple times under different rates of convergence. The lowest cost primer design from all simulated annealing runs is used as the final primer tiling design (Figure 2.3).

**Design results:** To test Ambre-design, we analyzed cell-line copy number data to identify a large clustering of deletions in the CDKN2A region (Greenman et al., 2010). We identified a 380kb region surrounding the CDKN2A gene, 230kb upstream and 150kb region downstream of CDKN2A that captures breakpoints in 55 of the 109 CDKN2A deletion cell-lines considered. We chose $d = 6500$, as 13kb products can be reliably amplified with LongAmp Taq DNA polymerase (New England Biolabs, NEB).

The candidate primer generation and primer filtering stages resulted in 5181 candidate primers. As shown in Figure 2.3a, the candidate primers are uniformly spread across breakpoint regions suggesting good tiling primer designs may exist. The simulated annealing algorithm is repeated for 12 different rates of convergence with the fastest convergence rate having a 10 minute average runtime and slowest convergence rate having a 864 minute average runtime (Figure 2.8). When $d = 6500$, the lowest cost solution (AMBRE-68) requires only 68 primers with 99.99% in silico capture of simple CDKN2A deletions that may occur in the 380kb breakpoint region (Figure 2.3b).
2.3.2 Sequencing amplified sequences harboring SVs

Sequencing the AmBre-amplify DNA confirms capture of CDKN2A deletions. We used PacBio RS technology due to its long reads, ideal for structural variation calling, and throughput, appropriate for medium sized experiments. Using computation, we correct for the high inherent error in PacBio sequencing.

Furthermore, if different samples do not share breakpoints (for example, all amplicons are of different sizes and amplify from different primer pairs within the design), the samples can be mixed and sequenced on a single run without additional barcoding. We employed this strategy with CDKN2A deleted samples on a single SMRT cell and relying on computation to deconvolute the breakpoints.

Define a breakpoint as a pair of disjoint coordinates $a$ and $b$ on a reference, and a non-template sequence $s$ (of length $\ell$) such that the sample sequence brings $a$ and $b$ together, separated only by the insertion of $s$. The objective of Ambre-analyze is to take as input a collection of PacBio sample sequences aligned to the reference genome and output a collection of breakpoints along with the sequence around each breakpoint. The code for this tool is stand-alone and can be used in the analysis of PacBio reads for SV detection. Ambre-analyze works by (a) alignment trimming (defined below), (b) breakpoint clustering of fragments, and (c) consensus sequence generation around each breakpoint (Figure 2.2, see Methods).

**Alignment trimming:** Denote a local alignment (Chaisson and Tesler, 2012) as a pair of intervals from the fragment and reference that can be aligned with a small number of edits. A split mapped fragment $F$ supports a breakpoint $(a, b, s)$ with two local-alignments (denoted as $(F_a, G_a), (F_b, G_b)$). In the ideal case, $G_a$ ends at $a$, and $G_b$ begins at $b$, while the fragment segment between $F_a$ and $F_b$ is exactly the inserted sequence $s$ (Methods). However, in real data, a fragment
can span multiple breakpoints, sequence errors can result in spurious incorrect alignments, and the alignments output by standard tools like BLASR (basic local alignment with successive refinement) will have inaccurate boundaries. Specifically, inaccurate boundaries might result in overlapping consecutive segments $F_a, F_b$. AmBre-analyze resolves these errors by choosing the optimal alignment segments covering the fragment $F$. For a fragment $F$, the input is a chain of local alignments $\mathcal{F} = (F_a, G_a), (F_b, G_b), \ldots$. The output is a subset $\mathcal{F}' = (F_a', G_a'), \ldots$ of $\mathcal{F}$, with alignment boundaries trimmed so (1) none of the fragment segments $F_a', F_b', \ldots$ overlap, (2) the number of distinct alignments is minimized, and (3) most of fragment $F$ is covered. The second and third objectives reinforce the notion that a typical fragment covers a small number of breakpoints and is mostly well-aligned except for non-template insertion sequence. The first objective helps to narrow down the breakpoint coordinates. To clarify, consider a trimmed reference interval $G_a'$ that ends at $x$ and a consecutive interval $G_b'$ beginning at $y$, while the gap between corresponding fragment segments is $L$. Then, we expect that $a > x$, $b < y$, and

$$L \simeq \ell + (a - x) + (y - b)$$

Thus, the fragment constrains the location of the breakpoint $(a, b)$ to lie in a small region between $x, y$. In the next section, we will use information from multiple fragments to further narrow the breakpoint location. Given these three distinct objectives, the alignment trimming algorithm works by combining them into a single objective function, and uses a dynamic programming approach to identify the optimal trimming (Methods).

**Fragment clustering:** Consider a two dimensional representation of the genomic space with $F$ and $R$ being the vertical and horizontal axes, respectively. In this
representation, a true breakpoint \((a, b)\) is represented by a point, and each split-mapped read \((x, y, L)\) is represented by a triangle of possible breakpoints \((a, b)\) that satisfy \((a - x) + (y - b) \leq L\) (Methods). Multiple reads supporting the same breakpoint represent multiple triangles whose intersection reduces the uncertainty in breakpoint determination. Furthermore, if reads from multiple AmBre-amplify experiments are combined, the split-mapped reads will cluster according to overlap, revealing breakpoints for each experiment sample. We develop a fast, customized method to recover the aggregated read clusters for each breakpoint (Methods). The method took 2.5 min seconds on a single desktop core to analyze all local alignments from 52,000 reads from a single PacBio SMRT cell experiment.

**Consensus sequence determination:** Predicted amplicon sequences are generated from the breakpoint estimates. In turn, these templates are supplied as reference sequences into PacBio’s SMRT Analysis Resequencing Protocol. The analysis protocol calls consensus amplicon sequences by correcting the predicted templates.

### 2.3.3 Identifying \textit{CDKN2A} deletion given DNA break clustering

AmBre exploits the fact that variable breakpoints aggregate along fragile regions of the chromosome by designing primers around the fragile regions. We used this idea to produce a single design for five cancer cell lines: A549, CEM, Detroit562, MCF7, and T98G. Breakpoints were estimated by copy number changes for four cancer cell-lines (A549, CEM, MCF7, and T98G) from SNP-array data (Greenman et al., 2010) (Table 2.1 and Figure 2.9) and the breakpoint was given for a fifth cell line (Detroit562) from prior studies. The error in breakpoint estimation for SNP-
array data is roughly 10kb. Thus, to generate cluster target regions, each breakpoint estimate was expanded to be a 10kb interval and overlapping intervals were merged. This created four regions ($F$) upstream of $CDKN2A$ and three downstream regions ($R$), and the target regions were used as input for AmBre-design ($d = 6500$bp). AmBre-design output a high quality 16 primer design (AMBRE-16) with primers spaced apart by approximately 6kb to cover the 100kb input region. The design was used by AmBre-amplify on DNA samples from each cell line. The experiment successfully amplified DNA from each cell line (Figure 2.10), where each line produced a unique sized amplicon even though each reaction uses the same set of 16 primers.

PCR products were mixed together for simultaneous preparation and sequencing on a single SMRT cell. The sequence data was the input to AmBre-analyze. The tool BLASR (Chaisson and Tesler, 2012) identified 52k alignable fragments. After clustering in AmBre-analyze, we retrieved deep coverage of every breakpoint (although with six clusters instead of five; see below), with A549 having the lowest coverage of 400 fragments and CEM having the highest coverage of 18,000 fragments (Figure 2.4). The difference in coverage is due to different amplicon sizes, where shorter amplicons are easier to load onto a PacBio SMRT cell than longer amplicons. Newer PacBio instrumentation is expected to normalize for this sequencing bias (Mason and Elemento, 2012).

AmBre-analyze generated consensus sequence for each cell line. A549, CEM, and Detroit562 breakpoints (Figure 2.11-2.12) are concordant with previous studies (Kitagawa et al., 2002; Sasaki et al., 2003; Bashir et al., 2009). The A549 harbors a complex structural variation where in addition to a large DNA segmental loss including $CDKN2A$, there is a 325bp internal inversion occurring at the deletion breakpoint junction. AmBre-analyze resolved the complex event as two separate
breakpoints. The A549 amplicon template was created by ordering the reference segments corresponding to the two breakpoints. After template refinement, the A549 amplicon sequence matched the sequence found by (Bashir et al., 2009).

To our knowledge, the nucleotide sequence for MCF7 and T98G had not been previously characterized in spite of previous efforts, including whole genome sequencing of the MCF7 cell line. The ease of the discovery in our experiment attests to the value of a targeted approach to SV detection. Both MCF7 and T98G sequences were confirmed using Sanger sequencing. Interestingly, the SNP-array estimate for MCF7 breakpoint is 15kb away from the AmBre detected breakpoint. The difference may be due to SINE and LINE repeats that mark the region of the upstream MCF7 breakpoint, a fact confirmed by the Sanger reads (Figure 2.11). Repetitive sequences are known to confound structural variation analysis and possibly explains why previous genome sequencing studies of MCF7, have not annotated the CDKN2A deletion breakpoints (Hampton et al. 2009; 2011).

We analyzed the physical properties of DNA around the breakpoints of CDKN2A deletions using the BreakSeq pipeline (Lam et al., 2009). All five deletion events were predicted to result from non-homologous end joining (NHEJ). According to (Lam et al., 2009), a characteristic of NHEJ is lower DNA duplex stability near the breakpoints of a structural variation. They assessed DNA duplex stability based on predictions of helix stability (average dissociation free energy of overlapping dinucleotides) and DNA flexibility (average twist angle of overlapping dinucleotides). We found no strong association to lower DNA duplex stability in CDKN2A deletion breakpoints, albeit we are analyzing much fewer structural variations (Figure 2.13). Alternatively, (Kitagawa et al., 2002) suggested that the CDKN2A deletion in CEM is due to illegitimate V(D)J recombination, which is evidenced by V(D)J recombination motifs discovered near the deletion breakpoints.
2.3.4 Characterizing CDKN2A deletion assuming no DNA break clustering

Also, AmBre applies to contiguous break regions. We developed a 68 primer design to capture CDKN2A deletions with breaks in a 380kb region (AMBRE-68, Figure 2.3).

In AmBre-amplify experimentation, we observed that the high amount of multiplexing, and larger amplicon lengths (> 4kb) reduce amplification efficiency. Using all AMBRE-68 primers in a single reaction resulted in amplification of only the 2.2kb A549 CDKN2A deletion loss (data not shown). To mitigate this effect, sub-sampling of primers from a design and performing multiple reactions per sample using different primer sets improved amplification results. To test whether the AMBRE-68 primers selected were viable at some level of subsampling, we sampled the nearest forward and reverse primer in AMBRE-68 to each CDKN2A break in cell lines: A549, CEM, Detroit562, MCF7, MOLT4, T98G. This resulted in a nine primer subset, which again captures the CDKN2A deletion in each cell line. Of these cell lines, five lines resulted in amplicons ranging in lengths from 2.2kb to 7.5kb (Figure 2.5). The Detroit562 breakpoints did not fall within the target breakpoint region given to AmBre-design and the expected amplicon size using the closest AMBRE-68 primers is 16kb. Thus, Detroit562 did not amplify with the nine primer subset. For each remaining cell line, the observed amplicon length matched the spacing between CDKN2A breakpoints and nearest primers in AMBRE-68 design. Thus, a universal primer design divided into multiple primer subset experiments can be used to identify SVs.
2.3.5 Characterizing \textit{RUNX1}-\textit{RUNX1T1} translocations

AmBre also captures more complex rearrangements like interchromosomal translocations. This was demonstrated with an experiment characterizing \textit{RUNX1}-\textit{RUNX1T1} gene fusion, the result of a translocation between chr21 and chr8. In the tumor genome, breakpoint ends lie within a 30kb region \textit{chr}21 : 36,205,000 – 36,235,000 in the \textit{RUNX1} intron, and a 55kb region \textit{chr}8 : 93,030,000 – 93,085,000 in \textit{RUNX1T1}, and the derivative chromosome 8 (Der8) encodes a fusion oncoprotein. In some cases, the translocation is balanced and also generates a fusion of \textit{RUNX1T1}-\textit{RUNX1} on a derivative chromosome 21 (Der21). To capture the translocation producing Der8, we used AmBre to design 10 reverse primers in the \textit{RUNX1} region and 18 forward primers in the \textit{RUNX1T1} region with \sim3kb primer spacing. Similarly, to capture Der21 breakpoints, 10 forward and 19 reverse primers were designed in the \textit{RUNX1} and \textit{RUNX1T1} regions, respectively. Recall, a \sim3kb primer spacing supposes the maximum product size is approximately 6kb. The primer designs were tested on Kasumi-1, which carries the balanced translocation with both Der8 and Der21 breakpoints characterized (Xiao et al., 2001). AmBre spaced the primers in the two regions unaware of the true Kasumi-1 breakpoints and we assayed the Der8 and Der21 chromosomes in two independent reactions using the respective 28 and 29 primers. The primers closest to the breakpoints produces a 3.5kb and 2.7kb amplicon from Der8 and Der21, respectively (Figure 2.6). Both reactions resulted in a strong signal and virtually no background noise, despite there being close to 30 primers in each reaction.

Furthermore, we investigated subsampling of primers and efficacy in generating longer amplicons. For each primer design, we divided the forward and reverse primers based on index parity when sorted by chromosome position. Thus, there are four primer sets: forward odd (FO), forward even (FE), reverse odd (RO), and
reverse even (RE), with primers spaced by approximately 6kb. The forward and reverse primer sets make four combinations: FO ∪ RO, FO ∪ RE, FE ∪ RO, and FE ∪ RE, primers for capturing target breakpoints. These combinations can be treated as four new primer designs, each with a maximum product size of 12kb, but half as many primers. This gives us the opportunity to assess amplification efficiency across different amplicon lengths and primer density per reaction using the same DNA template. In the original 28 primer design, the Kasumi-1 breakpoints for Der8 were generated by the sixth forward and ninth reverse primer. Thus, trying the 14 primer designs FE ∪ RO, FO ∪ RO, and FO ∪ RE produces 3.5kb, 6.8kb, and 10.1kb amplicons (Figure 2.6). Similarly, the 29 primer design for Der21 was subsampled into three reactions. Each reaction resulted in a strong signal band at the expected amplicon size and all six amplicon were confirmed to span the Der8 and Der21 breakpoint via Sanger sequencing (Figure 2.9.9). From each reaction, a general trend of better amplification for shorter amplicon lengths is observed. However, there was no significant difference in amplification efficiency between using all primers and half the primers to generate the shortest amplicons. Longer amplicons had strong signal, but weaker false products were visible. This effect is not seen with the shorter amplicons and false products may be more prevalent in reactions with greater number of primers and longer amplicons.

2.3.6 Dealing with tumor heterogeneity

The AmBre assay, unlike other methods, can target DNA with a SV in the context of high background of germline DNA. This feature is important for sensitive detection of tumor DNA and establishing a patient specific tumor DNA marker for monitoring tumor burden. We successfully amplified a 2.2kb CDKN2A deletion sequence from A549 and a 3.6kb deletion sequence from MCF7 starting
with A549 and MCF7 genomic DNA mixed with HEK genomic DNA (Figure 2.14). Each reaction starts with a heterogeneous mixture of approximately 400ng with tumor to wild-type gDNA mixture ratios of 1:1, 1:10, 1:100, 1:1000. In a realistic application for AmBre, each reaction contains numerous primers where only 2 primers are responsible for amplification. In the experiment, each reaction contains 16 primers sampled from AMBRE-68 around CDKN2A deletion breakpoints for each cell line. In the heterogeneity experiment of A549, strong amplification is observed for each mixture ratio whereas for MCF7 there is clearly a reduction of amplification efficiency as the fraction of starting cancer cell line gDNA decreases (Figure 2.14). Amplification of longer amplicons with AmBre in the complex gDNA sample is also possible, however with reduced sensitivity (Figure 2.9.11). The sensitivity for the AmBre assay is largely dependent on expected amplicon length. CDKN2A deletion breakpoints corresponding to a smaller amplicon in a particular AmBre primer design are more easily amplified.

2.4 Discussion

AmBre addresses the challenge of highly sensitive SV targeting in complex DNA mixtures. This is accomplished with a careful design of tiling primers that enables amplification of DNA harboring the SV if present in the mixture and a specialized PacBio analysis pipeline to confirm SV breakpoints. AmBre was used to discover breakpoints associated with CDKN2A deletion in cancer cell-lines MCF7 and T98G. In addition, we demonstrated amplification occurs even in a complex DNA mixture where 1 in every 1000 DNA molecules contain the CDKN2A deletion. These features of AmBre are clinically important. A SV breakpoint specific to a cancer patient could serve as a personalized biomarker, where a quantitative
PCR assay could accurately measure the patient’s tumor burden (Michor et al., 2005; Bartley et al., 2010). With advancements in microfluidics and droplet PCR, quantifying 1 – 3 copies of tumor DNA in a complex sample is possible (Hatch et al., 2011).

If the problem is to simply observe a SV, there are numerous high-throughput methods; SNP hybridization arrays (SNP-array), whole exome sequencing (WES), and whole genome sequencing (WGS). However, these methods are not ideal for a clinical application in tumor burden monitoring. SNP-arrays and WES give copy number read outs of DNA, which hint at the presence of SVs and a low resolution estimate of corresponding breakpoints. Without a high accuracy breakpoint estimate, a quantitative PCR assay specific to tumor DNA cannot be designed. WGS is capable of breakpoint calling, but would require an exorbitant amount of deep sequencing to capture SVs occurring in a low fraction of DNA. (Harismendy et al., 2011) reported the extent of this sequencing challenge, where more than 1500X coverage of cancer mutational hotspots (71.1kb region) was necessary to capture single nucleotide variants (SNVs) occurring with prevalence greater than 5% in the sample.

Therefore, a targeted approach for mutation detection is preferred to a high throughput untargeted mutation discovery for clinical practice. A high throughput method captures numerous SVs and SNVs where follow-up functional analysis is required for each mutation to determine its potential as a cancer driver or passenger mutation. Alternatively, there are numerous targetable SVs known to drive cancer progression, and are being used in clinical laboratories to confirm cancer diagnosis and guide therapy. The most notable example, CML patients with the $BCR-ABL1$ translocation are treated with tyrosine kinase inhibitors. The patient’s response to therapy can be reliably tracked by measuring tumor DNA containing $BCR-$
ABL1 gene fusion from blood samples (Michor et al., 2005; Bartley et al., 2010). Unfortunately, such success in tumor burden monitoring has not been observed for patients with solid tumors.

In this work, we present AmBre’s application to capture RUNX1-RUNX1T1 translocations in AML cases and CDKN2A deletions, which are prevalent in many types of cancer. Using the accompanying software, this approach can be easily extended to target other SVs, like BCR-ABL1 in chronic myeloid leukemia, EML4-ALK in lung cancer, and TMPRSS2-ERG in prostate cancer. For EML4-ALK and TMPRSS2-ERG, DNA breaks within introns and rearrangement of the chromosome fuse the genes together, similar to RUNX1-RUNX1T1 gene fusion. The remaining challenge for AmBre is a limited targetable breakpoint region. We presented a design capturing breakpoints falling within a 100kb and proposed a multiple primer subset strategy for encompassing a 380kb breakpoint region. Further development is necessary to capture SVs with breakpoints appearing in a greater than 1mb range. AmBre is a first step to sensitive tumor DNA monitoring test for solid tumors. Extending the approach with improvements of applying multiple primer designs to target the same SV or the use of microfluidic devices may lead to an ultra-sensitive assay capable of minimally invasive early cancer detection.

2.5 Methods

2.5.1 AmBre: Primer generation and filtering

Primer3 2.3.0 (Rozen and Skaletsky, 2000) was used with long-range PCR specific parameters to identify 31bp candidate AmBre primers that were capable of amplification under the same thermocycling conditions. To minimize the chance of off-target amplification, candidate primers were aligned to the reference human
assembly (GRCh37) using BLAT (Kent, 2002). Define an *end-aligning* match as an exact match of length > 18 between the 3′ end of a primer and an off-target location. Primers with greater than 10 end-alignments were removed as having a high chance for off-target amplification. Second, pairs of primers that have compatible end-alignments within a 2d long off-target region were marked as incompatible. Finally, each pair (including a self-pair) was tested for dimerization using MultiPlx (Kaplinski et al., 2005). Primers with self-dimerization (maximum binding energy $\Delta G$ less than $-8.0$ kcal/mol for any region) were removed and pairs with high binding affinity (maximum binding energy $\Delta G$ less than $-4.0$ kcal/mol for primer-primer 3end binding or $-8.0$ kcal/mol for any region of primers) were marked as incompatible. The remaining candidate primers and incompatibilities formed the input to AmBre primer selection.

### 2.5.2 AmBre: Primer selection with simulated annealing

A final AmBre primer design was selected from a filtered list of candidate primers ($P_U$) and primer-primer compatibilities. To compute an optimal primer design, a low cost $P$ according to $C(P)$, we applied a simulated annealing (Kirkpatrick, 1984) procedure. We computed an initial design $P$ using a random subset of 6 primers. Define the neighboring design of $P$, $N(P)$, as either the removal of a single primer from $P$, or the addition of a single primer $p \not\in P$ to $P$ followed by removal of all primers $p' \in P$ s.t. $(p, p') \in E$. The simulated annealing procedure described in Algo 1 was used to compute low cost designs.

The temperature schedule, $T_1, T_2, T_3, \ldots$, linearly decreases depending on intercept and slope parameters $m$ and $b$. Parameters tested for $T$ were combinations of $m = 1, 0.1, 0.01, 0.001$ and $b = 10^4, 10^5, 10^6$. The maximum number of iterations ran was determined by the temperature schedule, $2b + \frac{b}{m}$, and constrained to be
Algorithm 1 Simulated Annealing Algorithm

1: procedure SIMULATEDANNEALING($P_U, C$)
2: \[ P \leftarrow \text{Random}(P_U, 6) \] \hspace{1cm} \triangleright \text{Initialize random primer set } P \text{ with size 6}
3: for \( t = T_1, T_2, T_3, \ldots \) do \hspace{1cm} \triangleright \text{Iterate until design is stable}
4: \[ l \leftarrow \text{Random}(P_U, 1) \]
5: if \( C(N_l(P)) < C(P) \) or Random[0, 1] < \( e^{-\frac{C(N_l(P)) - C(P)}{t}} \) then \hspace{1cm} \triangleright \text{Move to neighboring design if improves or with probability proportional to extra cost and iteration}
6: \[ P \leftarrow N_l(P) \]
7: end if
8: end for
9: return \( P \)
10: end procedure

at least \( 10^6 \) and at most \( 10^8 \) iterations. Each parameter set was repeated 3 times. The lowest cost primer design of all runs was used as the final design. Figure 2.8 demonstrates convergence to design minima under different parameters of \( T \) for a target CDKN2A breakpoint region of length 380kb.

2.5.3 AmBre-analyze: PacBio sequence analysis

Alignment trimming: BLASR computed local alignments between the PacBio reads and human reference assembly were provided as input to alignment trimming. An alignment pair \((F_a, G_a), (F_b, G_b)\) with \( a \ll b \) between a fragment \( F \) and reference \( G \) imply a breakpoint. The goal of alignment trimming is to trim the ends of each alignment for each fragment \( F \), so that (a) each segment of \( F \) participates in a single alignment; and, (b) \( F \) is maximally covered.

We first remove local alignments encompassed by other alignments (e.g., 4 in Figure 2.7). We sort remaining alignments by their location on the fragment, so that alignment \( i \) starts before alignment \( j \) if and only if \( i < j \). Let \( b_s(i) \) and \( b_e(i) \) denote the fragment breaks before the beginning and after the end of alignment \( i \).

We represent alignments on a grid with alignments as rows and fragment
positions as columns (Figure 2.7). An alignment is a series of breaks on the fragment (i.e. \((1, b_1)\) to \((1, b_5)\) in Figure 2.7). Alignments are chained together to cover a portion of \(F\) exactly once. To chain adjacent alignments, for each alignment \(j\) with an alignment \(i\) that terminates before \(j\) starts, add a jump from \((i, b_e(i))\) to \((j, b_s(j))\) (for instance \((1, b_e(1))\) to \((3, b_s(3))\)). Also, for each alignment \(j\) overlapping an earlier alignment \(i\) on the fragment, add a jump from \((i, b_s(j))\) to \((j, b_e(i))\) (for instance \((2, b_e(3))\) to \((3, b_s(2))\)) if \(i\) spans \(b_s(j)\) and \(j\) spans \(b_e(i)\). By this process, any alignment chain covers positions exactly once.

\[
 w[(i, u), (j, v)] = \begin{cases} 
 \text{Aln}[i, u, v] & \text{if } i = j \\
 \frac{1}{2} (\text{Aln}[i, u, v] + \text{Aln}[j, u, v]) + J(u, v) & \text{if } i \neq j \text{ and } i, j \text{ overlap from } u \text{ to } v \\
 J(u, v) & \text{Otherwise.}
\end{cases}
\]

An alignment chain is scored by summing local alignment scores (\(\text{Aln}[i, u, v]\) for alignment \(i\) for fragment coordinates \(u\) to \(v\)) and penalizing for jumps between alignments (\(J(u, v)\) for alignment \(u\) to \(v\)). A high scoring alignment chain corresponds to trimmed alignments that aligns well and covers most of the fragment. The score of a chain is computed using dynamic programming. Let \(S(j, v)\) denote the score of the best chain ending at \((j, v)\). Then,

\[
 S(j, v) = \max_{(i, u)} \{S(i, u) + w[(i, u), (j, v)]\} \quad (2.2)
\]

In the recursion, \((i, u)\) is the start of alignment \(j\), start of a jump to \((j, v)\) (i.e. if \((j, v) = (3, b_e(2))\) then \((i, u)\) could be \((2, b_s(3))\), or previous position on alignment \(j\) where a jump ends (i.e. if \((j, v) = (2, b_e(2))\) then \((i, u) = (2, b_e(1))\)).
By not computing the score for each alignment and fragment position on the grid, the optimal trimmed alignment chain is quickly found.

Along the maximum scoring chain, each jump, \((F'_a, G'_a), (F'_b, G'_b)\), represents a breakpoint estimate \((a, b, F'_j - F'_i)\). For example, the jump from 1 to 3 corresponds with breakpoint estimate \((x_1, y_2, 6)\).

In this formulation, two alignments that overlap may contribute to a high score since the overlap segment is scored as the average of both alignment scores. Above, for a breakpoint estimate from overlapping alignments, we use boundaries around the overlap and do not resolve a tighter breakpoint within the overlap segment. Finding a tighter breakpoint estimate would require computing \(S\) for all breaks within overlap intervals, which is inefficient for thousands of fragments. In any case, the conservative breakpoint estimates are improved with downstream clustering and refinement steps.

**Breakpoint clustering:** Breakpoint estimates from all fragments supporting the same breakpoint are aggregated into groups using a sweep line algorithm. (Sindi et al., 2009), applied a similar geometric approach to efficiently identify structural variations using discordant paired end reads.

For a breakpoint estimate \((x, y, L)\), the true breakpoint junctions \((a, b)\) in reference \(G\) lies between \(x \ldots x + L\) and \(y - L \ldots y\), respectively, subject to \(a - x + y - b < L\). Here, we assume \(L\), a spacing length on \(F\), is a reasonable estimate for breakpoint uncertainty on \(G\) and the effect of sequencing deletion errors at the breakpoint junction is minimal. On a \(G \times G\) plane, each breakpoint estimate \(x, y,\) and \(L\) with the above constraints defines a triangle which contains the true breakpoint \((a, b)\) (Figure 2.7 and Figure 2.4).

A line sweeps the plane and tracks when breakpoint triangles overlap along
the sweep line. Here, a cluster is a collection of triangles where each triangle overlaps one or more triangles in the cluster. The consensus breakpoint \((a, b)\) for the cluster is the mode of \((x, y)\) estimates (see Figure 2.4).

**Accounting for reverse orientation alignments:** With a slight modification, we can account for alignments in the reverse complement orientation to capture structural variations with inversions and bidirectional PacBio reads. PacBio reads DNA amplicons in both directions, in particular, read in the forward direction produces an alignment chain \((F_x, G_x), (F_y, G_y)\) and in the reverse direction \((H_y, RC(G_y)), (H_x, RC(G_x))\) where \(RC\) reverse complements the sequence \(G\). This is resolved by relabeling reverse complement alignments by a \(-\), such that \(H\) supports the breakpoint \((-y, -x)\).

The relabeling applies naturally to the sequence analysis pipeline. Alignment-trimming relies only on projections on sequenced fragments and therefore does not change. Each DNA amplicon containing a breakpoint is associated with two breakpoint estimates, \((x, y)\) generated from forward reading and \((-y, -x)\) from reverse reading.

In addition, the constraints of \(-y, -x, L\) in relation to \(-a, -b\) remain the same, therefore both forward and reverse direction breakpoint estimates have the same triangle orientation on the \(G \times G\) plane. All forward and reverse breakpoints are simultaneously recovered with the sweep line algorithm.

Using reverse complement alignments, breakpoints associated with inversions, like A549, are captured. In this case, a breakpoint corresponds with \((-x, y)\) and \((-y, x)\) or \((x, -y)\) and \((y, -x)\).

**Breakpoint reconstruction:** In the final step, predicted amplicon templates for each cluster are created by joining reference sequence \(G(6500 - a, a)\) and
The PacBio SMRT Analysis 1.4 pipeline for Resequencing is performed to refine the amplicon template predictions using all fragments generated from the SMRT cell (Figure 2.12). The Resequencing protocol involves running BLASR for mapping followed by Quiver for consensus sequence calling. The protocol accurately recovered the sequence around breakpoints; the consensus amplicon sequence starting at aligned $25 - a$ and ending at $b + 25$ matched either sequencing from previous studies or independent Sanger sequencing chromatogram (Figure 2.11). For clusters with $L > 0$, adding $L$ “N” nucleotides at the breakpoint junction of the predicted amplicon template had no effect on PacBio Resequencing protocol. In both cases, the correct amplicon breakpoint junction sequence was found.

### 2.5.4 Experimental Methods

A549, CEM, Detroit562, and T98G cells were thawed from Moore’s Cancer Biorepository. MCF7, HeLa, and HEK (293T) cells were collected from Rosenfeld Lab. Standard DNAzol protocol was used for DNA extraction and DNA was quantified with NanoDrop 2000 spectrophotometer. DNA products are visualized on 1% agarose gels with EtBr. Gel images are either color value inverted or color curve adjusted uniformly across the image for visual enhancement. All PCRs were performed on a BioRad iCycler instrument.

All PCR experiments used the following thermocycling conditions; initial denaturation at 95°C for 3 min, 10 cycles at 94°C for 20 sec, 64°C for 30 sec, 66°C for 15 min, 28 cycles at 94°C for 5 sec, 64°C for 30 sec, 66°C for 15 min + 20 sec for each cycle, final extension at 64°C for 45 min, and 4°C hold.
AMBRE-16 experiment

See supplemental materials for primer sequences. Standard protocol for NEB Crimson LongAmp Taq is used for 50µl PCR reactions with the following changes. The same mix of 16 primers was used in each reaction where each primer is present with final concentration of 0.2µM. Starting genomic DNA for each cell line reaction is 10ng. QIAquick PCR purification kit was used to clean up PCR samples. Samples were quantified and 2µg of A549 reaction sample was mixed with 1µg of each remaining cell line reaction sample and submitted for PacBio sequencing at the UCSD BioGem Core facility. Loading of DNA samples onto a PacBio SMRT cell is biased towards sequencing smaller amplicons and increasing the amount of A549 reaction sample containing an 11kb DNA fragment was necessary to sufficiently sequence the A549 DNA fragment.

AMBRE-68 experiment

See supplemental materials for primer sequences. Standard protocol for NEB Crimson LongAmp Taq is used for 50µl PCR reactions with the following changes. The same mix of 9 primer was used in each reaction where each primer is present with final concentration of 0.4µM. Starting genomic DNA for each cell line reaction is 20ng.

RUNX1-RUNX1T1 experiment

See supplemental materials for primer sequences. Standard protocol for NEB Crimson LongAmp Taq is used for 25µl PCR reactions with the following changes. All primer at 0.4 µM PCR experiments were under the conditions; initial denaturation at 95°C for 1 min, 10 cycles at 94°C for 20 sec, 63°C for 30 sec, 68°C for 2 min, 28 cycles at 94°C for 5 sec, 61°C for 30 sec, 66°C for 2 min + 5
sec for each cycle, final extension at 64°C for 30 min, and 4° hold. Subsampling experiments used the same primer concentration and thermocycling conditions except extension times for the first phase is 7 min and the second phase is 7 min with 10 sec increase per cycle.

**Tumor:Wild-type genomic DNA heterogeneity experiment**

See supplemental materials for primer sequences. Standard protocol for NEB Crimson LongAmp Taq is used for 50µl PCR reactions with the following changes. Each primer has final concentration 0.4µM. Each reaction contains ≈ 400ng gDNA, with the following tumor to normal DNA ratios: 200ng : 200ng, 40ng : 400ng, 4ng : 400ng, 0.4ng : 400ng. Normal DNA is derived from HEK cells.

**MCF7 and T98G PCR validation**

Primer pair sequences were generated using Primer3 2.3.0 given short genomic sequence around the MCF7 and T98G breakpoints as determined by PacBio sequencing and analysis. See supplemental materials for primer sequences. Standard protocol for NEB Standard Taq is used for 50µl PCR reactions starting with 250ng of genomic DNA.

### 2.6 Data Access

The sequencing data have been deposited at the NCBI Sequence Read Archive (SRA; http://www.ncbi.nlm.nih.gov/sra) under accession number SRX353044. The AmBre software is available at http://bix.ucsd.edu/AmBre.
2.7 Tables

Table 2.1: Five cell-lines with CDKN2A deletion breakpoints in GRCh37. Estimated breakpoints are according to CGP (Greenman et al., 2010). CGP coordinates were converted from NCBI36 to GRCh37 using UCSC liftOver (Hinrichs et al., 2006). The break coordinates for Detroit562 were identical to (Bashir et al., 2009), and the cell-line was not examined by CGP.

<table>
<thead>
<tr>
<th>Cell Line</th>
<th>Type</th>
<th>Estimated Breaks</th>
<th>Our Breaks</th>
<th>True Deletion Size</th>
<th>True Deletion Size Difference in Breaks</th>
</tr>
</thead>
<tbody>
<tr>
<td>A549</td>
<td>adenocarcinoma</td>
<td>21833542 - 22121634</td>
<td>21832459 - 22123318</td>
<td>288092</td>
<td>1083 - 1684</td>
</tr>
<tr>
<td>CEM</td>
<td>lymphoblastic leukemia</td>
<td>21828110 - 21992808</td>
<td>21828685 - 21996997</td>
<td>168887</td>
<td>575 - 4189</td>
</tr>
<tr>
<td>CEM</td>
<td>pharynx carcinoma</td>
<td>21828110 - 21992808</td>
<td>21828685 - 21996997</td>
<td>168887</td>
<td>575 - 4189</td>
</tr>
<tr>
<td>Detroit562*</td>
<td>pharynx carcinoma</td>
<td>21970804 - 21985229</td>
<td>21970804 - 21985229</td>
<td>14425</td>
<td></td>
</tr>
<tr>
<td>MCF7</td>
<td>breast carcinoma</td>
<td>21834611 - 21989073</td>
<td>21819532 - 21989621</td>
<td>154462</td>
<td>15079 - 548</td>
</tr>
<tr>
<td>T98G</td>
<td>glioblastoma</td>
<td>21868909 - 21991923</td>
<td>21865639 - 21992514</td>
<td>129014</td>
<td>3270 - 590</td>
</tr>
</tbody>
</table>
2.8 Figures

**Figure 2.1:** PAMP tiling design for capture of *CDKN2A* deletions. *CDKN2A* upstream and downstream breakpoint regions are defined on a germline genome, blue and red lines, respectively. Tiled forward primers (blue arrows) and reverse primers (red arrows) are spaced ≈1kb apart (width of hashed boxes) (not to scale with reference). Overlap of blue box and red box on tumor DNA represents a forward and reverse primer pair are less than 2kb apart and will lead to amplification of tumor DNA harboring *CDKN2A* deletion breakpoints.
AmBre pipeline

AmBre-design: Primer designing
1) Candidate primer generation - Primer3 2.3.0
2) Primer filtering and primer-primer compatibility checking
3) Primer tiling selection optimization problem solved by simulated annealing

AmBre-amplify: experiments
1) PCR assay
2) Multi-sample PacBio preparation and single molecule sequencing

AmBre-analyze: Breakpoint sequence analysis
1) Sequence fragment mapping - BLASR
2) Alignment filtering and trimming.
3) Geometry-based fragment clustering and breakpoint estimation
4) Amplicon template sequence refinement

Figure 2.2: AmBre pipeline with primer designing and PacBio long fragment sequence analysis.

Figure 2.3: Designing AMBRE-68
A) Candidate primers are uniformly distributed in CDKN2A locus suggesting good primer designs are possible. AmBre-design is tasked to capture CDKN2A deletion upstream and downstream breakpoints in regions chr9 : 21,730,000 – 21,965,000 and chr9 : 21,975,000 – 22,129,000 (GRCh37 coordinates), respectively. B) Final low cost 68-primer design to capture CDKN2A deletions in 380kb breakpoint region. The solution has a 97.6% and a 99.7% coverage of breakpoint regions. The fraction of break pairs captured by the design (resulting in amplicon length < 13kb) is 99.99%.
Figure 2.4: Aggregates of breakpoints from each PacBio fragments after sweep line clustering. Target amplicons are strongly supported by fragments and breakpoints are well separated. Only breakpoints with $L < 1$kb are displayed for inset boxes. The height of each cluster corresponds with number of fragments supporting the breakpoint (depth of breakpoint coverage).
Figure 2.5: Subsampling of 9 primers from the complete AMBRE-68 tiling design results in clean amplification of CDKN2A loss DNA fragments in six cell lines. From left to right, lanes contain 1kb Plus GeneRuler DNA ladder, PCR products from samples A549 (2.2kb), CEM (5.8kb), MCF7 (3.6kb), MOLT4 (6.8kb), T98G (7.5kb), HEK, and water. The expected lengths of each amplicon according to AMBRE-68 design are listed in parentheses. HEK cells (no CDKN2A deletion) and $H_2O$ are negative controls.
Figure 2.6: Characterizing $RUNX1$-$RUNX1T1$ balanced translocation in Kasumi-1. Lanes 1,2,4,6 and 8 contain 1kb Plus GeneRuler DNA ladder, PCR products from Kasumi-1 Der8 with all 28 primers (3.5kb), 14 primer $FE \cup RO$ (3.5kb), 14 primer $FO \cup RO$ (6.8kb), 14 primer $FO \cup RE$ (10.1kb). Lanes 3,5,7 and 9 contain matching water controls, which show no contamination. Lanes 10,12,14, and 16 contain PCR products from Kasumi-1 Der21 with all 29 primers (2.7kb), 15 primer $FO \cup RO$ (2.7kb), 15 primer $FE \cup RO$ (6.1kb), and 14 $FE \cup RE$ (8.1kb). Gel was loaded with $2\mu l$ for lanes 2,3,4,5,10,11,12 and 13, and $4\mu l$ for remaining volumes. Reactions with shorter amplicons amplified extremely well and lesser volumes were used for visualization on the gel. The expected amplicon lengths according to the Der8 and Der21 design are listed in parenthesis.
**Figure 2.7**: A) Fragment-segmentation example for local alignments 1, 2, 3, and 4 along a PacBio fragment. B) Triangle representation of adjacent alignments 1, 2, and 3 on $G \times G$ plane.
2.9  AmBre supplement

2.9.1 Comparison to Bashir et al. cost function

The cost function, $C(P)$ differs from Bashir et al. (2007), $B(P)$. Unlike $C(P)$, $B(P)$ penalizes for loss of break region coverage independent from limiting too many primers in a design. More specifically, $B(P)$ applies a cost weight to the sum of too large primer spacings and limits the number of primers by adding the cost $w_G \frac{|P|d}{T}$. If $\frac{|P|d}{T}$ is greater than primer density parameter $\rho$, then $B(P) = \inf$ by setting $w_G = \inf$, otherwise $w_G = 0$. While $B(P)$ works well for large regions requiring numerous primers, it does not work well for smaller regions with sparser selection of primers.

Consider a one-sided case where the input is one region of length $T$ in $F$ and $R$ is an empty set. For $1 \leq \rho < 1 + \frac{d}{T}$ and a primer set $(P)$ with $\lceil \frac{T}{d} \rceil$ primers, the cost of adding any compatible primer to $P$ without primer removal for $B(N_l(P))$ is $\inf$ whereas for $C(N_l(P))$ is definite. This becomes an issue when requiring sparse primer selection. An extreme example with any primer density $1 \leq \rho < 1.5$ where $T = 2d$, the simulated annealing procedure using $B$ would limit to only primer subsets of size 0, 1, and 2. However, setting $\rho = 1.5$ would have a definite cost for primer subsets of size $P$. Using $B$, how to set $\rho$ with sparse selection of primers in multiple regions is also not trivial. The reformulated cost function $C$, has the additional property where $C_{\rho_i}(N_l(P)) > C_{\rho_j}(N_l(P))$ for $\rho_i < \rho_j$, therefore changing $\rho$ has a reasonable effect on the new cost of primer subsets.

Also, under sparse primer selection, $B$ does not necessarily evenly space primers. Consider again the above one-sided case. Let $N = \frac{\rho T}{d}$ (number of primers desired) and $N_e = N - \lfloor \frac{T}{d} \rfloor$ (number of extra primers than necessary to cover $T$) and suppose each position of $T$ has a fully compatible candidate primer. The
ideal primer subset \( (P_I) \) would select a primer every \( \frac{T}{N} \) positions of \( T \). Consider a contrived primer subset \( (P_C) \) where a primer is selected every \( d \) positions of \( T \) and \( \frac{d}{N_e+1} \) positions of \( T \), which places primers to cover \( T \) entirely and places all extra primers within the first covered segment of \( T \). Note the cost \( B(P_I) \) equals the cost \( B(P_C) \) even though \( P_I \) is clearly better. The cost \( C(P_I) \) is zero and less than cost \( C(P_C) \) if there are extra primers to be placed, \( N_e > 0 \). This scenario is unlikely to occur with a large region and low primer density, however may occur with sparse primer selection and high primer density. Nonetheless, \( C \) reports sensible costs for both situations.

It follows from above, low-cost solutions with unevenly spaced primers can be generated by setting a primer density threshold for \( B(P) \) too high. For discontiguous target regions, in particular, some target regions may receive more extra primers than should be placed in that region. Our cost function penalizes for irregular primer spacing, thus each target region will receive the appropriate number of primers.
2.9.2 Figure for simulated annealing convergence

Figure 2.8: Designing AMBRE-68 Simulated annealing using different convergence rates, is used to select good primer designs with lowest cost. The convergence rate that finds the lowest cost primer design will depend on the input given to AmBre-design.

2.9.3 AmBre designs with breakpoint estimates from CGP

Figure 2.9: Breakpoint estimates for A549, CEM, MCF7, and T98G from CGP. Last two rows are AMBRE-16 and AMBRE-68 input target regions.
2.9.4 Figure for AMBRE-16 amplifications

**Figure 2.10**: PCR products of AMBRE-16 on cell-lines: A549 (lane 2), CEM (lane 3), Detroit562 (lane 4), HeLa (lane 5), MCF7 (lane 6), and T98G (lane 7). 4µl of 1kb GeneRuler in lane 1. Lanes are reactions starting with 10ng cell-line genomic DNA. HeLa cells (no CDKN2A deletion reported by CGP) and H2O are negative controls. Arrow denotes weak Detroit562 band; another PCR had stronger amplification and was used for subsequent sequencing.

2.9.5 Breakpoint sequences

**Figure 2.11**: Breakpoint sequences for A549, CEM, Detroit562, MCF7, and T98G with orthogonal validation chromatogram of MCF7 and T98G. AmBre-analyze captures both breakpoints and non-templated insert sequence (highlighted in yellow).
2.9.6 PacBio coverage of refined amplicon sequences

Figure 2.12: BLASR remapping to consensus amplicon templates. The GC content for A549, CEM, Detroit562, MCF7, and T98G amplicons is 0.40, 0.42, 0.42, 0.38, and 0.38, respectively.
### 2.9.7 Primer sequences for AMBRE-16 and AMBRE-68

<table>
<thead>
<tr>
<th>AMBRE-16 primer sequences</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2-21815419-T-r0.2-3</td>
<td>TTTCTCTCTTAGATTGGAATAATTGGTGGAAC</td>
</tr>
<tr>
<td>A2-21818824-T-r0.2-3</td>
<td>TACGTTGTCTATTAGCTATAATCACCATGCAG</td>
</tr>
<tr>
<td>A2-21826327-T-r0.2-3</td>
<td>TTAATAAGCCTTCTAGCTGGAAGATTCACCAC</td>
</tr>
<tr>
<td>A2-21833470-T-r0.2-3</td>
<td>TCACCTCTCTGTGTATTAGACGACAGAAATTG</td>
</tr>
<tr>
<td>A2-21861703-T-r0.2-3</td>
<td>ATGGCAATAAGTGATTACGAAACAAATGCTC</td>
</tr>
<tr>
<td>A2-21867239-T-r0.2-3</td>
<td>GTTACTCTTGCTTTATTCTCAACAGCAGAGG</td>
</tr>
<tr>
<td>A2-21879976-T-r0.2-3</td>
<td>TTTCACTGAGCGAAAATCTAAGGATTATGTG</td>
</tr>
<tr>
<td>A2-21885329-T-r0.2-3</td>
<td>TCTTAAGGGTGGCCAGGATTACTATAACC</td>
</tr>
<tr>
<td>A2-21891029-T-r0.2-3</td>
<td>TAGAACCGGTAGTTTAGAAACAACTGTCTAG</td>
</tr>
<tr>
<td>A2-21967542-T-r0.2-3</td>
<td>CCCCTAGAGTTTCTATTCTCATATTATAACC</td>
</tr>
<tr>
<td>A2-21987671-F-r0.2-3</td>
<td>TATTATGACCCCTTTGTATGAATTGGAAAAG</td>
</tr>
<tr>
<td>A2-21993806-F-r0.2-3</td>
<td>ACACACAGTAGAAAGGGTGATTTCAAGGC</td>
</tr>
<tr>
<td>A2-22001902-F-r0.2-3</td>
<td>ACAAGGACTTGACTGGAGATAGAGACCTAG</td>
</tr>
<tr>
<td>A2-22036684-F-r0.2-3</td>
<td>AATGGACAGATGACTCTAACTTTGAATTAG</td>
</tr>
<tr>
<td>A2-22121823-F-r0.2-3</td>
<td>AGTTATAGAGATCCAGGCATCCAAACAC</td>
</tr>
<tr>
<td>A2-22128237-F-r0.2-3</td>
<td>GTAGCAACAGCAGTAGCAGTATAACAGCAAC</td>
</tr>
</tbody>
</table>
9 primer sequences used from AMBRE-68 design

<table>
<thead>
<tr>
<th>Chr</th>
<th>Primer Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr9-21817993-T</td>
<td>TACTCATCACGGGTAAACATTTCTTCTCTC</td>
</tr>
<tr>
<td>chr9-21825345-T</td>
<td>GATGTCTTTTCTTGACGTAAAAATTTCATTTCC</td>
</tr>
<tr>
<td>chr9-21831236-T</td>
<td>AATCCTGATGATTGTAGGAAATCAGTACACC</td>
</tr>
<tr>
<td>chr9-21861845-T</td>
<td>TTTATTTAAAGTGATGTATGTTTCCTGCGTCCTC</td>
</tr>
<tr>
<td>chr9-21960424-T</td>
<td>CACTTTCTGACTGCACTTTTCTTAAGTATTTC</td>
</tr>
<tr>
<td>chr9-21991669-F</td>
<td>CCACGTAGTTCCATCATCAACTAACCATG</td>
</tr>
<tr>
<td>chr9-21991669-F</td>
<td>CCACGTAGTTCCATCATCAACTAACCATG</td>
</tr>
<tr>
<td>chr9-21996284-F</td>
<td>GTAAGAGTGATCGCTAAATCTCCTAATTTTTTCC</td>
</tr>
<tr>
<td>chr9-22124303-F</td>
<td>TACATTGTCTACCACCTCCCTTTCTTAATG</td>
</tr>
</tbody>
</table>

MCF7 and T98G validation primer sequences

Primer sequences used for MCF7 validation

<table>
<thead>
<tr>
<th>Forward</th>
<th>Primer Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAAGAGGCCCTGTTGTG</td>
<td></td>
</tr>
<tr>
<td>GGGGTTAGTGACTCGAGAC</td>
<td></td>
</tr>
</tbody>
</table>

Primer sequences used for T98G validation

<table>
<thead>
<tr>
<th>Forward</th>
<th>Primer Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCAGCAAGGCCAAAGAATCTGA</td>
<td></td>
</tr>
<tr>
<td>TTAGCCACTGTGACCGGTAA</td>
<td></td>
</tr>
</tbody>
</table>
2.9.8 DNA helix stability around breakpoints

![Graph showing DNA helix stability around breakpoints for different cell lines.]

Figure 2.13: Using code from BreakSeq pipeline, DNA flexibility for the 6 breaks around proposed non-homologous end joining DNA breaks showed no significant deviation.

2.9.9 \textit{RUNX1-RUNX1T1} translocation Sanger confirmation

Subsampled positive PCRs from Kasumi-1 were sent to GENEWIZ for sequencing after Qiagen PCR clean-up. Each sample sequenced with a forward primer upstream and a reverse primer downstream of the expected Kasumi-1 breakpoint. Samples A, B and C are shortest to longest amplicons from Der8. Similarly, samples D, E and F are amplicons from Der21.

<table>
<thead>
<tr>
<th>BLASTN 2.2.28+</th>
</tr>
</thead>
</table>

\textbf{Query=} KASUMI Derivative chromosome 8 AML-ETO

\textbf{Length=240}

\begin{tabular}{lll}
<table>
<thead>
<tr>
<th>Sequences producing significant alignments:</th>
<th>Score</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>lcl|8236 A=ai-1_kas_T_A08.ab1</td>
<td>422</td>
<td>2e-122</td>
</tr>
<tr>
<td>lcl|8237 B=ai-1_kas_T_B08.ab1</td>
<td>444</td>
<td>5e-129</td>
</tr>
<tr>
<td>lcl|8238 C=ai-1_kas_T_C08.ab1</td>
<td>346</td>
<td>1e-99</td>
</tr>
<tr>
<td>lcl|8239 A=ai-1_kas_F_D08.ab1</td>
<td>255</td>
<td>2e-72</td>
</tr>
<tr>
<td>lcl|8240 B=ai-1_kas_F_E08.ab1</td>
<td>298</td>
<td>4e-85</td>
</tr>
</tbody>
</table>
\end{tabular}

\textbf{ALIGNMENTS}

\begin{tabular}{ll}
<table>
<thead>
<tr>
<th>BREAKPOINT</th>
<th>Score</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 91 GACCCGTAGTTCTCTTATGAGTAAGAAAGCTTGAGAACACTTTCCTGTATATTTAGA</td>
<td>180</td>
<td></td>
</tr>
<tr>
<td>8236 192</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>8237 183</td>
<td>94</td>
<td></td>
</tr>
</tbody>
</table>
\end{tabular}
Query= KASUMI Derivative chromosome 21 ETO-AML

Sequences producing significant alignments:

<table>
<thead>
<tr>
<th>Accession</th>
<th>Description</th>
<th>Score (Bits)</th>
<th>E Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>lcl</td>
<td>19454</td>
<td>D-a2-2_kas_T_G08.ab1</td>
<td>438</td>
</tr>
<tr>
<td>lcl</td>
<td>19455</td>
<td>E-a2-2_kas_T_H08.ab1</td>
<td>438</td>
</tr>
<tr>
<td>lcl</td>
<td>19456</td>
<td>F-a2-2_kas_T_A09.ab1</td>
<td>438</td>
</tr>
<tr>
<td>lcl</td>
<td>19457</td>
<td>D-a2-2_kas_F_B09.ab1</td>
<td>285</td>
</tr>
<tr>
<td>lcl</td>
<td>19458</td>
<td>E-a2-2_kas_F_C09.ab1</td>
<td>285</td>
</tr>
<tr>
<td>lcl</td>
<td>19459</td>
<td>F-a2-2_kas_F_D09.ab1</td>
<td>292</td>
</tr>
</tbody>
</table>

Primer sequences for Kasumi-1 breakpoints

Der8 primer sequences for 3.5Kbp

<table>
<thead>
<tr>
<th>Primer</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>d2.1-6-T</td>
<td>AGTACACTAGAGCACCATAAGAATAACATCC</td>
</tr>
<tr>
<td>d2.1-9-F</td>
<td>CAATACCTCGCTACTCAAAGCTTTCT</td>
</tr>
</tbody>
</table>

Der8 primer sequences for 6.8Kbp

<table>
<thead>
<tr>
<th>Primer</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>d2.1-6-T</td>
<td>AGTACACTAGAGCACCATAAGAATAACATCC</td>
</tr>
<tr>
<td>d2.1-10-F</td>
<td>CCAGTAGGAAGACAGTCATGTGAAGATG</td>
</tr>
</tbody>
</table>

Der8 primer sequences for 10.1Kbp

<table>
<thead>
<tr>
<th>Primer</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>d2.1-5-T</td>
<td>CAAGCCAAAATACCACAATCATCCTAAGAC</td>
</tr>
<tr>
<td>d2.1-10-F</td>
<td>CCAGTAGGAAGACAGTCATGTGAAGATG</td>
</tr>
</tbody>
</table>

Der8 primer sequences validation

<table>
<thead>
<tr>
<th>Primer</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>kas-a1.1-T</td>
<td>TGCGAGAAATCAAACAAACATTG</td>
</tr>
<tr>
<td>kas-a1.1-F</td>
<td>TTTTCTGATCGAGACTTTTCTGCCC</td>
</tr>
</tbody>
</table>
Der21 primer sequences for 2.7Kbp

<table>
<thead>
<tr>
<th>Primer</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1.1-9-T</td>
<td>TTTGCCCTCTATAATAGACAGTCTTCAG</td>
</tr>
<tr>
<td>d1.1-13-F</td>
<td>CTTCCCTAGGACTTTGGTTTTAATGATTC</td>
</tr>
</tbody>
</table>

Der21 primer sequences for 6.1Kbp

<table>
<thead>
<tr>
<th>Primer</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1.1-8-T</td>
<td>AAGAGCAAAGAGTGACACATCTTTTCATC</td>
</tr>
<tr>
<td>d1.1-13-F</td>
<td>CTTCCCTAGGACTTTGGTTTTAATGATTC</td>
</tr>
</tbody>
</table>

Der21 primer sequences for 8.5Kbp

<table>
<thead>
<tr>
<th>Primer</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1.1-8-T</td>
<td>AAGAGCAAAGAGTGACACATCTTTTCATC</td>
</tr>
<tr>
<td>d1.1-14-F</td>
<td>CAGTCTTACTTGTGTCTTTTACTTTCCAGTCC</td>
</tr>
</tbody>
</table>

Der21 primer sequences validation

<table>
<thead>
<tr>
<th>Primer</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>kas-a2.2-T</td>
<td>TTCCACGCATTTAGTTTTCCCAC</td>
</tr>
<tr>
<td>kas-a2.2-F</td>
<td>CTGCCCTGCTAATTTCCAGGAT</td>
</tr>
</tbody>
</table>

2.9.10 Amplification in complex gDNA samples with AMBRE-68 on A549-HEK and MCF7-HEK

![Figure 2.14](image)

Figure 2.14: Successful A549 (red arrow) and MCF7 (green arrow) CDKN2A deletion amplification with heterogeneity ratios 1 : 1, 1 : 10, 1 : 100, 1 : 1000 (lanes 3-6 for A549 and lanes 10-13 for MCF7) and 16 primers starting with 400ng of gDNA. Lane 1 contains 1kb Plus Gene Ruler DNA ladder. Lanes 2 and 9 are A549 and MCF7 positive control reactions starting with 20ng of homogenous gDNA. Lanes 7,14 are negative control reactions with wild-type DNA and lanes 8,15 are water negative control reactions with corresponding 16 primer mixes.
2.9.11 Amplification in complex gDNA samples with longer PCR products and lower multiplexing

**Figure 2.15**: A549 and Detroit562 with 6 primers and heterogeneity ratios. Starting with a total 250ng of DNA material, we demonstrate amplification of a 7.6kbp CDKN2A deletion sequence from A549 and a 9.7kbp deletion sequence from Detroit562 with 6 primers and tumor to wildtype DNA mixtures of 1:1,1:10, and 1:100 (Fig. 2.15). Standard protocol for NEB Crimson LongAmp Taq is used for 25µl PCR reactions with the following changes. Each primer has final concentration 1.1µM. Each reaction contains ≈ 250ng of DNA, with the following tumor to normal DNA ratios: 125ng : 125ng, 25ng : 250ng, 2.5ng : 250ng, 0.25ng : 250ng. Normal DNA is derived from HEK cells.

**2.10 Acknowledgements**

Chapter 2 is adapted from Patel A, Schwab R, Liu Y, Bafna V. Amplification and thrifty single-molecule sequencing of recurrent somatic structural variations. Genome Research. (2013). The dissertation author was the primary author of this paper, and was responsible for the research.
Chapter 3

The elusive evidence for chromothripsis.

3.1 Abstract

The chromothripsis hypothesis suggests an extraordinary one-step catastrophic genomic event allowing a chromosome to “shatter into many pieces” and reassemble into a functioning chromosome. Recent efforts have aimed to detect chromothripsis by looking for a genomic signature, characterized by a large number of breakpoints (50 to 250), but a limited number of oscillating copy number states (2 to 3) confined to a few chromosomes. The chromothripsis phenomenon has become widely reported in different cancers, but using inconsistent and sometimes relaxed criteria for determining rearrangements occur simultaneously rather than progressively. We revisit the original simulation approach and show the signature is not clearly exceptional, and can be explained using only progressive rearrangements. For example, 3.9% of progressively simulated chromosomes with 50 to 55 breakpoints were dominated by two or three copy number states. In addition,
by adjusting the parameters of the simulation, the proposed footprint appears more frequently. Lastly, we provide an algorithm to find a sequence of progressive rearrangements that explains all observed breakpoints from a proposed chromothripsis chromosome. Thus, the proposed signature cannot be considered sufficient proof for this extraordinary hypothesis. Great caution should be exercised when labeling complex rearrangements as chromothripsis from genome hybridization and sequencing experiments.

3.2 Introduction

In a groundbreaking study 2011 study Stephens et al. (2011), Stephens et al. observed a pattern of structural variation in a leukemia genome so atypical it presumptively revealed a novel mechanism of chromosome rearrangement. Two features distinguish this variation pattern. First, the chromosome or chromosomal region in question has many clustered breakpoints that suggest complex adjacencies rather than simple deletions or non-overlapping tandem duplications. Second, the region oscillates between two or perhaps three copy number states.

To further investigate this phenomenon, Stephens et al. sequenced several cell lines with chromosomes that exhibited these features. One of these chromosomes was chromosome 15 from SNU-C1, a colon cancer cell line. This chromosome has 239 breakpoints identified by paired-end sequencing (PES) and mostly oscillates between two copy number states, two and four. Using simulations, Stephens et al. showed that the progressive introduction of the breakpoints they observed would result in a chromosome with many copy number states rather than just two. They hypothesized that the peculiar rearrangement pattern was not the result of progressive rearrangements but instead the result of the chromosome shattering
followed by the random stitching together of the resulting pieces. They termed this phenomenon “chromothripsis”.

To determine how widespread chromothripsis may be, Stephens et al. used the progressive rearrangement simulation from SNU-C1 to conclude that a chromosome with at least 50 breakpoints dominated by at most three copy number states was unlikely to have been rearranged progressively and thus was likely to be a product of chromothripsis. Using these criteria they searched copy number profiles and estimate 2-3% of cancers have a chromosome that bears the hallmark of chromothripsis.

This is a striking result; it suggests a mechanism of cancer genome evolution that contrasts starkly with previously described models. This discovery has generated excitement and ongoing investigation. Subsequent studies have found evidence for chromothripsis in multiple myeloma Magrangeas et al. (2011), medulloblastoma Rausch et al. (2012); Northcott et al. (2012), neuroblastoma Molenaar et al. (2012), and colorectal cancers Kloosterman et al. (2011) as well as the germline Kloosterman et al. (2011); Chiang et al. (2012a). Moreover in some studies, chromothripsis has been associated with more aggressive cancers. Thus, it would appear that a new source of human disease has been found, with potentially far-reaching effects on our understanding and treatment of cancer Patel et al. (2011).

The great potential of chromothripsis cannot be realized unless it can be accurately detected. It is unlikely that chromothripsis will ever be reliably observed directly, so we will need to rely on the footprint that chromothripsis should leave in copy number and breakpoint data. The characterization of this footprint is an open problem. While Stephens et al. searched for chromosomes dominated by at most three copy number states with at least 50 positions where copy number changes, subsequent works have used more relaxed criteria. They have required
fewer breakpoints per chromosome, such as 20 Molenaar et al. (2012), 10 Rausch et al. (2012); Northcott et al. (2012), or just a handful Chiang et al. (2012a). They also have not always required that the number of unique copy states in a chromosome be limited to two or three Northcott et al. (2012); Molenaar et al. (2012); Korbel and Campbell (2013).

The validity of these footprints of chromothripsis rests on the idea that progressive rearrangement cannot create such patterns. However, the evidence for this proposition is largely limited to the initial simulation work by Stephens. Chromothripsis is now being investigated in different contexts than Stephens’ cell line simulations. Furthermore, the diversity of approaches used to identify chromothripsis means some groups are likely over- or underestimating its prevalence. This, together with the potentially great significance of chromothripsis, highlights the value of revisiting and extending the simulation work that underlies current strategies for identifying chromothripsis.

In this article, we review the simulation approach that suggests that progressive rearrangements cannot yield a chromosome with many breakpoints and few unique copy number states. First, we explore whether changes to the implementation of the simulation affect the validity of the footprint of chromothripsis. We show that a subtle but consequential error in the original implementation of the simulation causes it to understate the breakpoint and copy number patterns that can be achieved by progressive rearrangement. We examine varying possible meanings of “breakpoint” and “copy number state” and determine definitions that more closely correspond to experimental results. Next, we show that progressive rearrangement with a preference for inversions can produce chromosomes that bear the putative footprint of chromothripsis. Together these issues suggest that, assuming the simulation approach is valid, more stringent criteria must be used to
identify chromothripsis and that the current literature overstates its prevalence.

We then demonstrate that the simulation approach produces similar results whether a chromosome is progressively rearranged or not. This undermines its ability to distinguish between chromothripsis and progressive rearrangement. Extending on this finding, we demonstrate a method that finds plausible progressive rearrangements that explain the breakpoints of particular chromosomes that have been documented to have undergone chromothripsis. Finally, we discuss the significance of these findings and question the chromothripsis hypothesis.

### 3.3 Methods

#### 3.3.1 Simulating Progressive Rearrangements

We will first summarize the simulation method. Consider a chromosome 100 bases long that undergoes chromothripsis, shattering into ten segments of ten bases, which we label a through j. The segments come back together, but some are lost, some are inverted, and the order is shuffled. Suppose the resulting arrangement of segments is ae(-c)(-g)(-h)j. If this chromosome is sequenced, it will reveal five breakpoints and copy numbers that alternate between zero and one. The breakpoints are shown in Table 3.1, and an illustration of breakpoint positions is shown in Figure 3.1a.

We can now step through the progressive rearrangement simulation used by Stephens et al. The simulated chromosome begins intact, with no rearrangements (Figure 3.2a). Then, a random breakpoint is chosen from the set of observed breakpoints. In this case, suppose the breakpoint between 60 and 80 is chosen. This breakpoint is now introduced into the chromosome via one of three rearrangement types: inversion, deletion, or tandem duplication. The observed orientation of
the two ends of the breakpoint is head-to-tail. So, an inversion cannot be used to create the breakpoint because that will result in segments with orientations of head-to-head or tail-to-tail. A deletion between 60 and 80 will not work because then the orientations would be tail-to-head. But, a tandem duplication between 60 and 80 will result in a breakpoint that, when read from 60 to 80, will join the head of segment \( g \) to the tail of segment \( h \). So, segments \( g \) and \( h \) are duplicated. Suppose, the breakpoint between 20 and 70 is chosen and segments \( c \) to the furthest segment \( g \) are duplicated. Next, the rearrangement between 30 and 50 is chosen. Using similar reasoning as above, a rearrangement that reproduces the breakpoint is introduced. Suppose the rearrangement chosen is an inversion of segments \( fghgc \). Note that this creates two breakpoints, the observed breakpoint plus another one in head-to-head orientation. Then, two more rearrangements are introduced resulting in the chromosome in Figure 3.2f. The number of breakpoints and copy number states in this chromosome would be recorded, and the simulation would be repeated many times with different rearrangement orders and segment choices. It would also be stopped when specific numbers of breakpoints had been introduced so that the relationship between the number of breakpoints and the number of unique copy number states could be determined.

3.3.2 Finding Chromosome Rearrangements Consistent with Observed Breakpoints

In addition to simulations, we can propose sequences of rearrangements that explain all the observed breakpoints. Each breakpoint is provided as a pair of coordinates that is non-adjacent in the reference genome, but adjacent in the donor sample. We first construct a breakpoint graph Bafna and Pevzner (1996) from breakpoints (e.g., Figure 3.1b graph is constructed from breakpoints
in Table 3.1): Partition the chromosome of length $L$ into $n$ segments based on breakpoint coordinates $p_1, p_2, \ldots$. Each segment has two ends labeled by their position on the chromosome with the 5′ segment marked as head ($h$) and the 3′ segment end marked as tail ($t$). The breakpoint graph is described by $2n$ nodes, denoted $\{(p_0, h), (p_1, t), (p_1, h), (p_2, t), \ldots, (p_{2n-1}, t), (p_{2n-1}, h), (p_{2n}, t)\}$ where $p_0 = 1$ and $p_{2n} = L$. Pairs of nodes are connected by segment-edges and breakpoint-edges. Segment-edges (shown in grey in Figure 3.1b) connect $(p_i-1, h)$ to $(p_i, t)$ for all $1 \leq i \leq 2n$. Thus, every node has exactly one segment edge. Recall that the input is a set of breakpoints, $\langle (p_i, e_i), (p_j, e_j) \rangle, \ldots$, where $p_i, p_j$ correspond to the chromosomal positions that are brought together, and $e_i, e_j$ are each either head or tail. By definition, the breakpoint graph already has the nodes $(p_i, e_i)$ and $(p_j, e_j)$, and we connect each such pair with a breakpoint-edge (shown in green). When constructing the graph from the Stephens et al. data, no node had more than one breakpoint edge. The observed loss of heterozygosity in chromosomes suggested that a single chromosome undergoes rearrangement. Our method uses the haploid assumption, by finding a single path in the breakpoint graph. Alternatively, finding two disjoint paths in the breakpoint graph corresponds to two rearranged chromosomes, without changing the observed breakpoints.

Based on sorting of reversals theory Bafna and Pevzner (1996), a continuous path from the start node to the end node reads out a sequence of alternating segment and breakpoint edges representing the rearranged chromosome. Given the continuous path, there is always a sequence of reversals (inversions) that transforms the rearranged chromosome back into the original segment ordering. In Figure 3.1b, the only graph component of this type is the path from $(1, h)$ to $(100, t)$. If the set of observed breakpoints is not complete, we will have multiple connected components. This is remedied by chaining paths together with new breakpoint edges. The
termini of paths with breakpoint edges are nodes with a single outgoing segment edge. To create a full path between chromosome start and end nodes, begin with a path containing the start node and connect the non-start terminus to the terminus of a randomly selected path by adding a new breakpoint edge. Continue the process of joining paths until all paths with observed breakpoint edges are consumed and finishing with the path containing the end node. With the continuous path as input, GRIMM Tesler (2002) finds a sequence of inversions transforming the original segment order into the rearranged chromosome. Note that once the breakpoint graph is created, all rearrangements impact only the breakpoint junctions between segments, and breakpoint reuse is allowed Alekseyev and Pevzner (2007).

Deleted segments are found in isolated connected components comprising of exactly two nodes and one segment edge. In Figure 3.1b, there are four deleted segments. The last possibility for connected components in the graph are cycles, which represent duplicated segments. For example, a simple cycle with a breakpoint edge connecting the head of a segment to its tail is a tandem duplication. To determine the order of rearrangements to create the observed chromosome, deletions are randomly placed in the inversion order and duplications are introduced last to preserve low copy number states.

3.4 Results

3.4.1 Simulating Progressive Rearrangements

Stephens et al. graciously shared the code they used to produce their results. We have reimplemented the method, applied it to chromosome 15 of SNU-C1, and replicated their results (Figure 3.2a). The general trend, consistent with Stephens’ result, is that the number of unique copy number states increases with the number
of breakpoints. A chromosome with 239 breakpoints and only two copy number states falls well outside of what was produced by the progressive simulation, and this is a key piece of evidence that chromosome 15 of SNU-C1 is the result of chromothripsis rather than progressive rearrangement. Moreover, based on the chart, it appears that a chromosome with at most three copy number states and more than fifty, or perhaps even twenty, breakpoints also falls outside of what can be achieved by progressive rearrangement.

3.4.2 Chromothripsis Footprint Criteria Depend on Subtle Simulation Implementation Details

The above result is more meaningful if it is robust to changes in the implementation of the simulation. In this section, we alter the simulation in various ways to determine if the proposed footprint of chromothripsis remains valid when assumptions about progressive rearrangement are changed.

The first change we made to the simulation was a correction of a logic error that caused some simulated inversions to behave like duplications. The details are in the supplement, but the net effect was that some operations that ought to have preserved existing copy numbers instead introduced up to two new copy number states to the chromosome. When we corrected this, the chart of copy number states and breakpoints shifted down (Figure 3.2b). This change in result does not affect inferences about chromosome 15 of SNU-C1, since 239 breakpoints and only two copy number states is still well outside of the simulated results. But, the simulated chromosomes now begin to encroach upon the chromothripsis region of the graph. For example, the new simulation produced a chromosome with 67 breakpoints and only 3 copy number states, which is consistent with the footprint of chromothripsis even though the chromosome was rearranged progressively.
The next alteration was to the counting of breakpoints and copy number states. Thus far, we have been imprecise about the meaning of the breakpoint values on the x-axes of our charts. This imprecision is also found in the literature, but there are in fact multiple ways to count breakpoints on a chromosome. One way is to count the number of times an abnormal adjacency appears. For example in the chromosome in Figure 3.2f, moving from left to right we find eight such adjacencies: e(a), e(-c), (-c)(-g), (-g)(-h), (-f)d, hg, g(-i), and (-h)j. This counting method was used in Figures 3.2a and 3.2b. Another way to count breakpoints is to consider how the breakpoints would be reported by a paired-end sequencing (PES) experiment Tuzun et al. (2005). This is similar to the previous method, except that if an abnormal adjacency appears in the chromosome multiple times because of duplications, it will only appear once in the sequencing results. So referring back to Figure 3.2f, the adjacencies hg and (-g)(-h) would count as one adjacency, even though they appear twice on the chromosome. A third way to count breakpoints is to consider how they will appear in a microarray or depth of coverage experiment Yoon et al. (2009). This method counts breakpoints where copy number changes. The copy numbers in the chromosome in Figure 3.2e from left to right are 1,0,1,2,4,3,1. So, copy number changes six times.

There are also multiple ways to count the number of copy number states in a chromosome. The first we can call “strict”. With this method, we simply count the number of copy number states observed in the chromosome, regardless of how much of the chromosome is covered by any copy number state. In Figure 3.2f, there are five copy states observed, zero through four. Another method, which we will call “relaxed”, counts how many copy states are needed to cover some fraction of the chromosome. If we use the fraction 90%, then the relaxed number of copy states in the chromosome above is four because we can cover 90 bases using only four copy
number states. Relaxed counting of copy states can be appropriate for identifying chromothripsis because it allows us to find chromosomes that are dominated by two or three copy number states but may have some small regions with other copy numbers because of subsequent alterations or experimental error.

The simulation by Stephens et al. used strict copy number state counting and the first breakpoint counting method, counting every unexpected adjacency even if duplicated. In contrast, the breakpoints observed in chromosome 15 of SNU-C1 come from PES, and the copy number state count of two was arrived at using relaxed counting. Microarray results show that the chromosome has six copy number states using strict counting Greenman et al. (2010).

We modified the simulation to use relaxed copy state counting that found how many copy number states were needed to cover 95% of the simulated chromosome. When this was combined with PES breakpoint counting, it produced the results in Figure 3.2c; when combined with microarray breakpoint counting, it produced Figure 3.2d. Because of the changes in breakpoint counting, the simulations could no longer quickly produce chromosomes with over 100 breakpoints. Both simulations also showed a continuation of the trend seen in Figure 3.2b with a narrowing separation between the simulated chromosomes and chromosomes bearing the footprint of chromothripsis. For example, of the 414 chromosomes in Figure 3.2c with between 50 and 55 breakpoints, 16 (3.9%) were dominated by three or two copy number states. This suggests that in a screen of many chromosomes, the proposed footprint of chromothripsis may produce false discoveries.

Finally, we altered the way the simulation chooses breakpoints to introduce into the chromosome. In the original simulation, breakpoints were chosen uniformly randomly without replacement, so each remaining breakpoint had an equal chance of being introduced at each step. This may not correspond to biological reality as
there may be some preference for particular kinds of rearrangements. Specifically, a preference for inversions over other rearrangement types could lead to chromosomes with many breakpoints but few copy number states. To test this, we changed the simulation so that inversions were twice as likely to be chosen at each step compared to deletions or duplications. The results are in Figures 3.3a and 3.3b, using PES and microarray breakpoint counting respectively. These results have many simulated chromosomes bearing the footprint of chromothripsis. The large fraction of chromosomes with many breakpoints and few copy number states (Table 3.2) indicates that some chromosomes that appear to have undergone chromothripsis could also have been produced by progressive rearrangement that favors inversions.

The results in this section suggest that a more conservative threshold should be used to identify chromothripsis in order to avoid false discoveries. If the minimum number of breakpoints were set at 100 rather than 50, much of the risk of false discovery we have demonstrated above would be diminished. However, this threshold would also decrease the estimate of the prevalence of chromothripsis. When Stephens et al. screened 746 cancer cell line copy number profiles for chromosomes with over 50 breakpoints and at most three copy number states, they found chromosomes from 18 cell lines that met these criteria. With a threshold of 100 breakpoints, the number of cell lines drops to 3. Based on this analysis, the true prevalence of chromothripsis may be less than .5% rather than the original estimate of 2-3%.

3.4.3 Simulation Method Does Not Distinguish Between Progressive Rearrangement and Chromothripsis

In the previous section, we discussed implementation details of simulations of progressive rearrangements. We now turn our attention to the question of whether such simulations can provide reliable evidence for chromothripsis at all. In order
for an experiment to provide information about a hypothesis, it has to produce different results when the hypothesis is true than when it is false. In order for simulations to demonstrate whether a chromosome could have been rearranged progressively, the simulations should produce different results for progressively rearranged chromosomes and chromosomes that have undergone chromothripsis.

The footprint of chromothripsis, many breakpoints with few unique copy states, is unlikely to appear in a chromosome rearranged by progressive and overlapping tandem duplications. However, it may appear in a chromosome rearranged by progressive inversions and deletions. We simulated such a chromosome with only inversions and deletions. The resulting breakpoints and copy numbers are shown in Supplemental Figure S3.7. The chromosome had 237 breakpoints and only two copy number states, zero and one. Even though only two kinds of rearrangements were used, the chromosome shows the same complex rearrangement pattern seen in chromosomes that have putatively undergone chromothripsis.

We then applied the simulation method to the breakpoints of this chromosome and recorded the results as we did in Figure 3.2b. The resulting distribution of breakpoints and copy number states in Figure 3.5 is not different from Figure 3.2b even though we know the chromosome was rearranged progressively. This result casts doubt on the usefulness of the simulation method to detect chromothripsis. Rather than distinguishing between chromosomes that shattered and chromosomes that were rearranged progressively, it always report that chromosomes with many complex rearrangements and few copy number states are the product of chromothripsis even when they are not.
3.4.4 Plausible Progressive Rearrangement Schemes Exist for Chromosomes Bearing Footprint of Chromothripsis

Thus far, we have discussed in general whether some chromosomes that appear to be the product of chromothripsis may actually have been progressively rearranged. We now move from the general to the specific to see if we can find series of progressive rearrangements that explain particular chromosomes that bear the footprint of chromothripsis. Stephens et al. singled out three chromosomes from three different cell lines for extensive sequencing and analysis: chromosome 5 from TK10, chromosome 9 from 8505C, and chromosome 15 from SNU-C1. These chromosomes had 55, 77, and 239 breakpoints respectively and oscillated between two copy number states. We developed a method that explains these breakpoints and copy number states using only progressive rearrangements.

As discussed above, one way to ensure that a chromosome has no more than two copy states is to only rearrange it by deletions and inversions. The problem of explaining genomic rearrangements using inversions alone, also known as the ‘Sorting by reversals’ problem, was solved by the Hannenhalli-Pevzner theory Hannenhalli and Pevzner (1995), and implemented in the tool GRIMM Tesler (2002). In this problem, the input is a pair of chromosomes with identical, but highly rearranged genomic content. The output is a sequence of inversions that transforms one chromosome to the other. For each of the three chromothripsis chromosomes, we identified arrangements of chromosomal segments that would yield the observed breakpoints using a graph traversal technique. In addition, some of the breakpoints support missing chromosomal segments. These missing segments were removed by progressively introducing deletions. Also, breakpoints supporting
tandem duplications are resolved last to ensure the fewest copy number states are observed. GRIMM then revealed inversions that would convert the unrearranged chromosome into one with the observed breakpoints (see Methods).

For each of the three chromosomes, we identified a sequence of inversions, deletions, and tandem duplications that yielded 100% of the experimentally observed breakpoints as well as some additional breakpoints beyond what was observed (Table 3.3). Figure 3.6 illustrates the result for chromosome 5 from TK10. Animations of the series of rearrangements for each of the three chromosomes are in the supplement.

These series of progressive rearrangements raise potential alternative hypotheses for the complex breakpoints and oscillating copy number states in these chromosomes. Thus, while these chromosomes may have indeed undergone chromothripsis, the observations can also be explained using progressive rearrangements alone.

3.4.5 The Alternative Explanation

Above, we have shown that simulations and the observed pattern of low copy number state count and high number of breakpoints clearly cannot distinguish chromothripsis from progressive rearrangements that favor inversions and deletions. However, we do not claim that a particular scheme of simple inversions and simple deletions causes the observed phenomenon. Inversions are the simplest generalization of a larger class of balanced rearrangements, which include translocations and rearrangements with multiple breakpoints. Specifically, define a \( k \)-break rearrangement, as an operation that rearranges \( k - 1 \) distinct segments of a chromosome creating \( k \) breakpoints. By this definition, an inversion is a specific type of 2-break rearrangement, while transpositions (and inverted transpositions)
are examples of 3-break rearrangements. Consistent with genome rearrangement theory Bafna and Pevzner (1996, 1998); Alekseyev and Pevzner (2008), a k-break rearrangement can be equivalently explained by a series of 2-break rearrangements (reviewed in Supplement Section 3). Using the k-break definition, chromothripsis can be described as an n-break rearrangement allowing for deletions. Our results show that a signature consisting only of breakpoints, cannot distinguish between progressive 2-break rearrangements and deletions from a one-off n-break rearrangement with deletions (chromothripsis). Since 2-break rearrangements decompose k-break rearrangements, progressive combinations of k-break rearrangements for $2 \leq k < n$ are equally plausible explanations for the observed breakpoints and copy number states. As each of these alternative scenarios provide a different number of rearrangement events, the number of rearrangement events cannot be accurately estimated using only breakpoint and copy number data.

### 3.5 Discussion

It is notoriously difficult to make sense of many cancer genomes due to the complexity of rearrangements. The proposal of the chromothripsis hypothesis was an important step forward as a possible mechanism for the creation of this complexity. Careful investigation of the phenomenon may deepen knowledge of structural variation in cancers.

At the same time, the proposal of ‘shattering and subsequent reassembly’ of a chromosome in a single catastrophic event is truly extraordinary. The invocation of chromothripsis to explain molecular data from cancer samples must be done with great circumspection, and caution, even. The case for chromothripsis rests on the argument that there are some patterns of variation that progressive rearrangement
cannot achieve. But in this paper, we have shown that progressive rearrangements can indeed achieve patterns that, at first glance, would seem quite unlikely. The primary evidence supporting chromothripsis Stephens et al. (2011) is (1) high breakpoint count and low copy number states. We demonstrated that this footprint of chromothripsis, in fact, includes chromosomes rearranged progressively, that simulations might always rule out progressive rearrangement regardless of how the chromosome truly evolved, and that it is possible to find progressive rearrangements that explain chromosomes that appear to be exemplars of chromothripsis.

Additional criteria used to argue for a chromothripsis event are: (2) clustering of breakpoint locations, (3) randomness of fragment joins, (4) rearrangements affecting a single haplotype, (5) interspersed loss and retention of heterozygosity, and (6) ability to walk the derivative chromosome Stephens et al. (2011); Korbel and Campbell (2013). However, these new criteria do not preclude chromosome formation via progressive rearrangements. For example, progressive rearrangements may produce the same pattern of (2) clustered breakpoint locations and (3) randomness of fragment joins. In our progressive rearrangement simulations, breakpoints were sampled from the exemplar chromosome 15 of SNU-C1, which has an identical distribution of breakpoint locations and breakpoint orientations. A recent review Korbel and Campbell (2013) reported patterns (4),(5), and (6), but did not provide quantitative analysis of these patterns against the few chromosomes proposed to have undergone chromothripsis. Pattern (4) suggests that breakpoints falling on one chromosome versus two homologous chromosomes is an indication of chromothripsis. Again, the pattern is indiscriminate since rearrangements may still progressively fall on a single chromosome. Additionally, patterns (5) and (6) are features specifically of rearrangements appearing on the same chromosome. (5) Interspersed loss and retention of heterozygosity occurs when segments are deleted from only one of the
homologous chromosomes. As the other homologous chromosome is intact, segments that are deleted appear to have loss of heterozygosity and remaining segments retain heterozygosity. The property of (6) walking the derivative chromosome states that a set of chromothripsis breakpoints when projected onto a reference chromosome, will allow for an unambiguous walk from one end of the reference chromosome to the other end of the chromosome traversing all the observed breakpoints. If both homologous chromosomes have rearrangements and since it is not known which breakpoints arise from which chromosome, finding an unambiguous walk is typically not possible. Along the reference there would be two walks and for each walk, inference would have to be made on which breakpoints to follow. However, if only one of the homologous chromosomes is rearranged, a single unambiguous walk is possible using all the breakpoints supporting the single rearranged chromosome.

In our analysis, like Stephens et al., we assume progressive rearrangements are occurring on the same chromosome and likewise the proposed patterns (4),(5), and (6) are inherently reproduced with progressive rearrangements in our simulations and plausible explanations of documented chromothripsis chromosomes.

In balance, our results suggest it is difficult to point to statistical evidence that predicts chromothripsis while excluding other possibilities. In this manuscript, we do not delve into biological explanations of the observed rearrangement patterns. For example, if rearrangements accumulate progressively, how could they be limited to a single chromosome. Isn’t a one-off catastrophic chromothripsis event a better explanation than some ‘memory’ that causes the same chromosome (or a few chromosomes) to be dramatically rearranged over time? However, there is evidence that chromosomal lesions might make a chromosome more susceptible to mutations. An example is provided by the breakage-fusion-bridge mechanism where the loss of a telomere and resulting instability leads to progressive cycles of rearrangements.
Indeed, Sorzano and colleagues propose the breakage-fusion-bridge and similar progressive mechanisms show patterns similar to chromothripsis Sánchez Sorzano et al. (2013). Chiang et al. (2012b) showed transgene integration in germline cells makes a chromosome more susceptible to rearrangements and the resulting complex rearrangements have chromothripsis-like patterns. While they suggest the rearrangements appear in a one-off event, we note that there are still numerous cell divisions between pronucleus injection of exogenous DNA and harvesting of DNA for sequencing. Based on this, we cannot rule out that rearrangements accrued over a few cell divisions. The rearrangements do not need to keep occurring in all cell divisions, as well. For example, breakage-fusion-bridge cycles occur across distinct cell divisions, but eventually lead to a stable rearranged chromosome. Similarly, Liu et al. (Liu et al. (2011), Table 1) provide numerous examples of complex rearrangements occurring on a single chromosome. They suggest that the results are best explained not as chromothripsis, but due to a “chromoanasynthesis”, as they involve errors in replicative mechanisms and show duplications and triplications not seen in chromothripsis. They provide the example of a patient with 18 copy number changes including a 5.5 Mbp triplicated and inverted segment. Breakpoint analysis revealed insertions of long (1.5Kbp) novel sequences at breakpoints, which might provide a template switch during the replicative process Liu et al. (2011). Once we admit these possibilities, however, the difference between one-off and progressive events becomes harder to measure.

We find purely statistical evidence cannot distinguish between one-off and progressive events (see also Righolt and Mai (2012); Sánchez Sorzano et al. (2013)). Other authors have noted that ‘chromothripsis like events’ cannot be distinguished from other complex genomic rearrangements Malhotra et al. (2013). While we use inversions and deletions to explain the chromothripsis patterns, we do not claim a
particular scheme of simple inversions and simple deletions causes the phenomenon. Inversions are the simplest generalization of a larger class of balanced rearrangements, which include translocations and rearrangements with multiple breakpoints (see Supplement Section 3). In other words, the chromothripsis pattern could also be explained by progressive steps of balanced rearrangements like translocations. Lastly, the appearance of balanced rearrangements and deletions is a plausible scenario for cancer. Wang et al. (2011) analyzed five T-ALL samples and observed 31 interchromosomal translocations, 19 intrachromosomal translocations, 1 inversions, 22 deletions, and 16 insertions. While they did not describe copy number changes, the number of breakpoints from possible copy neutral events is high ($31 + 19 + 1 \times 2 = 52$ breakpoints). Thus, it is reasonable to find a few extreme cancer chromosomes having higher number of balanced rearrangements and deletions.

### 3.6 Conclusion

These results do not foreclose upon the chromothripsis hypothesis, of course. But, they do underscore difficulty of making inferences about mechanisms in cancer. Indeed, there is no doubt that some of the cancer genomes have undergone extensive rearrangements. At the same time, the evidence is limited for the claim that a single catastrophic event joined shattered DNA together, and requires additional investigation before it can be accepted as established fact. For now, chromosomes with many breakpoints should be labeled as having undergone complex genome rearrangements (CGR) rather than implying a shattering mechanism by chromothripsis. Future advances in single-cell sequencing and haplotype resolved genome assembly might shed light on the mechanisms underlying complex rearrangements.
### 3.7 Tables

**Table 3.1:** Breakpoints of a rearranged chromosome in Figure 3.1.

<table>
<thead>
<tr>
<th>Lower Position</th>
<th>Lower Segment</th>
<th>Terminus of Lower Segment</th>
<th>Higher Position</th>
<th>Higher Segment</th>
<th>Terminus of Higher Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>tail</td>
<td>40</td>
<td>e</td>
<td>head</td>
</tr>
<tr>
<td>30</td>
<td>c</td>
<td>tail</td>
<td>50</td>
<td>e</td>
<td>tail</td>
</tr>
<tr>
<td>20</td>
<td>c</td>
<td>head</td>
<td>70</td>
<td>g</td>
<td>tail</td>
</tr>
<tr>
<td>60</td>
<td>g</td>
<td>head</td>
<td>80</td>
<td>h</td>
<td>tail</td>
</tr>
<tr>
<td>70</td>
<td>h</td>
<td>head</td>
<td>90</td>
<td>j</td>
<td>head</td>
</tr>
</tbody>
</table>

**Table 3.2:** Fraction of chromosomes in Figure 3.3a with few copy number states for given breakpoint counts.

<table>
<thead>
<tr>
<th>Breakpoint Range</th>
<th>Fraction of Chromosomes With 2 or 3 Copy Number States</th>
</tr>
</thead>
<tbody>
<tr>
<td>50-59</td>
<td>12.6%</td>
</tr>
<tr>
<td>60-69</td>
<td>7.4%</td>
</tr>
<tr>
<td>70-79</td>
<td>2.6%</td>
</tr>
<tr>
<td>80-89</td>
<td>0.8%</td>
</tr>
<tr>
<td>90-99</td>
<td>0.6%</td>
</tr>
</tbody>
</table>
Table 3.3: For each of the three chromothripsis chromosomes, the number of breakpoints observed experimentally, the number of unobserved breakpoints that were produced by the inversions and deletions, and the number of progressive rearrangement events broken down by rearrangement type.

<table>
<thead>
<tr>
<th>Cell Line</th>
<th>Experimental Breakpoints</th>
<th>New Breakpoints</th>
<th>Progressive Inversions</th>
<th>Progressive Deletions</th>
<th>Progressive Tandem Duplications</th>
</tr>
</thead>
<tbody>
<tr>
<td>TK10</td>
<td>55</td>
<td>12</td>
<td>63</td>
<td>42</td>
<td>3</td>
</tr>
<tr>
<td>8505C</td>
<td>77</td>
<td>21</td>
<td>91</td>
<td>56</td>
<td>1</td>
</tr>
<tr>
<td>SNU-C1</td>
<td>239</td>
<td>92</td>
<td>321</td>
<td>145</td>
<td>2</td>
</tr>
</tbody>
</table>
3.8 Figures

Figure 3.1: A hypothetical chromosome of length 100 shattered into blocks of length 10 and then reassembled. The hypothetical rearranged chromosome breakpoints are in Table 3.1. a) The breakpoints and block copy number projected onto original chromosome. b) A breakpoint graph representing the hypothetical rearranged chromosome with Table 3.1 breakpoints.
Figure 3.2: Following Stephens et al. simulation procedure, a sequence of possible rearrangements steps to explain the observed breakpoints in Table 3.1.
a: Results of directly reimplementing the simulation method of Stephens et al.

b: Results after fixing indexing issue for inversions.

c: Results counting breakpoints as they would appear from paired-end sequencing and counting the number of copy number states needed to cover 95% of the chromosome.

d: Results counting breakpoints as they would appear from microarrays or depth of coverage and counting the number of copy number states needed to cover 95% of the chromosome.

**Figure 3.3:** Charts of number of breakpoints versus number of copy number states for simulated chromosomes. The shaded gray area indicates the boundaries of the footprint of chromothripsis proposed by Stephens et al. The cell lines with number of breakpoints and copy number states as described by Stephens et al. are plotted as red points. The red dashed line shows the median number of copy number states for given numbers of breakpoints. The green dashed lines show an interval of copy number states that contains 99% of observations.
a: Result using paired-end sequencing  

b: Result using microarray breakpoint counting.

**Figure 3.4:** Charts of breakpoints versus copy number states for simulations with an overrepresentation of inversions.

**Figure 3.5:** Counts of breakpoints and copy number states from a simulation based on the breakpoints from simulated chromosome in Fig. S3.7. The breakpoints and copy number states of the simulated chromosome are indicated on the chart.
Figure 3.6: An illustration of the result of the series of inversions and deletions for chromosome 5 of TK10. The top panel broadly shows the ordering of segments after rearrangement. The upper color bar shows all segments of the unrearranged chromosome colored from blue to red. The lower color bar shows segments with the same coloring after rearrangement. Note that some segments have been deleted so the chromosome is shorter. The middle panel shows the breakpoints achieved by inversions and deletions, and the lower panel shows the observed breakpoints.
3.9 “The elusive evidence for chromothripsis.”

supplement

3.9.1 Indexing issue in the simulations from Stephens et al.

The simulations of progressive rearrangements were implemented in R by Stephens et al. The ordering of segments in the chromosome was maintained in a data frame called `curr.state`. Different types of rearrangements were effected by manipulating `curr.state`. A “head-to-head” inversion could be created with the following code:

```r
curr.state <- curr.state[c(1:left.seg, right.seg:(left.seg+1),
                         (right.seg+1):(dim(curr.state)[1]]),]
```

Here, `left.seg` and `right.seg` refer to the positions of the segments that are the boundaries of the region being inverted. If `left.seg ≤ right.seg`, the code correctly creates an inversion in `curr.state`. However, because of the varying orientations of segments in the simulated chromosome, it may be the case that `left.seg > right.seg`. If this is so, this code will triplicate `[right.seg+1,left.seg]` and will duplicate the segments at `right.seg` and `left.seg+1`. A similar issue exists for “tail to tail” inversions. This error causes the simulation to introduce two new copy number states when it should not introduce any.
3.9.2 Simulation of a chromosome simulated with progressive inversions and deletion

Figure 3.7: Breakpoints and copy number states of a chromosome simulated with progressive inversions and deletions. The number of breakpoints and copy number states is similar to SNU-C1. The breakpoints are colored according to orientation.
3.9.3 Equivalence of inversions, intra-chromosomal translocations, and inter-chromosomal translocations

Claim: A genome that is rearranged by one or more copy number neutral events (i.e. intra-chromosomal translocations) can be equivalently explained by a sequence of inversions.

Our manuscript suggests that a sequence of inversions suffice to explain the rearranged chromosome and copy number states of the cell-line SNU-C1, previously only explained by chromothripsis. This raises the natural question, is a sequence of inversions a better biological explanation? However, the question is misdirected. A more relevant question, are the other biological explanations as plausible as chromothripsis? We do not need to invoke inversions solely to explain the rearrangements. Here, we explain how any sequence of copy-neutral rearrangements can be equivalently explained by inversions. Conversely, a sequence of inversions can possibly be explained using a combination of copy-neutral rearrangements. To intuitively understand the equivalence, we consider three examples below.

- We show that an intra-chromosomal translocation is explained by a sequence of inversions.
- We define the concept of $k$-break rearrangements (also known as complex genome rearrangements), and show that both inversions and chromothripsis are special cases of $k$-break rearrangements.
- We extend these descriptions to inter-chromosomal translocations, and show that inter-chromosomal translocations can also be modeled as a sequence of inversions.
To reiterate, when we show that a complex rearrangement is modeled by a sequence of inversions, it is equivalent to saying that it is modeled by *some* sequence of copy neutral events. For more precise and thorough arguments of these ideas, see Bafna and Pevzner (1996); Alekseyev and Pevzner (2008); Bafna and Pevzner (1998).

**Intra-chromosomal translocation (transposition)**

Figure 3.8: Transposition of ‘de’ into ‘gh’.

The event of a DNA segment moving to a new position in the genome is referred to as transposition. An example of transposition is shown in Figure S3.8. In the example, the segment ‘de’ moves between ‘g’ and ‘h’, which rearranges “abcdefghij” to “abcfgdehij”. The removal of ‘de’ creates two breaks in sequence, and the insertion breaks the sequence ‘gh’. As a result, three breakpoint are observed, corresponding to the fusion of ‘cf’, ‘gd’, and ‘eh’. While the singular event is referred to as transposition, the breakpoints can also be explained by a sequence of three inversions, inversion of ‘de’, followed by inversion of ‘fg’, and finally inversion of ‘(-e)(-d)(-g)(-f)’ (Figure S3.8). Contrarily, when we say that
the rearrangements are explained by a sequence of inversions, we are only claiming that they are explained by some sequence of copy-neutral events.

Complex genome (or, $k$-break) rearrangements

![Complex Intrachromosomal Rearrangement Diagram]

Figure 3.9: Complex genome rearrangement with three breakpoints. Poly-fusion event in chromosome 8 involving three genes discovered by McPherson et al. (2012).

It is an intuitive but misguided conclusion that rearrangements leave characteristic signatures of paired-end discordant mappings. Certainly for simple, single events, this is true, and we see characteristic signatures for deletions, inversions, etc., as reported by many authors Wang et al. (2011). However, when we look at complex rearrangements, the signatures are no longer predictive. Transposition is an intuitive example of a one step multi-break rearrangement, and we showed in the previous example that it could be equivalently explained by inversions. To generalize this, Alekseyev and Pevzner Alekseyev and Pevzner (2008) developed the notion of $k$-break rearrangements and argued that as $k$ increases the number of
rearrangement events dropped dramatically. However, *given a rearranged genome, not much can be said about the type and number of rearrangements that created it.*

Here, we consider an empirical example. McPherson et al. (2012) analyzed cancer RNA-seq data and found numerous examples of aberrant gene fusions that implied complex genome rearrangement events. In Figure S3.9, we illustrate a poly-fusion, observed on chromosome 8 by McPherson et al. Within the tumor chromosome, the authors observed a poly-fusion of genic and non-genic parts comprised of inverted PHF20L1, an inverted non-genic section between PHF20L1 and FAM49B, inverted FAM49B, and SAMD12. In Figure S3.9, we illustrate a poly-fusion, observed on chromosome 8 by McPherson et al. Within the tumor chromosome, the authors observed a poly-fusion of genic and non-genic parts comprised of inverted PHF20L1, an inverted non-genic section between PHF20L1 and FAM49B, inverted FAM49B, and SAMD12. In Figure S3.9, chromosome 8 is represented by the sequence “abcdefghij” where segments represent genes as follows: ‘d’ corresponds to SAMD12, ‘e’ corresponds to FAM49B, ‘f’ corresponds to the non-genic section between PHF20L1 and FAM49B, and ‘gh’ is PHF20L1. The tumor chromosome 8 observed by McPherson et al. (2012) in the block representation is “abcd(-e)(-f)(-h)(-g)ij”. Since there was no evidence to support a particular DNA rearrangement mechanism, the authors refrained from suggesting how the complex event arose. We demonstrate two extreme scenarios for explaining the breakpoints observed in Figure S3.9. First, the breakpoints could be created in a single multi-break shuffle as shown in Figure S3.9. In the second scenario, the breakpoints could be explained by three sequential inversions, inversion of ‘e’, inversion of ‘f’, and inversion of ‘gh’. *Both scenarios equally explain the rearranged tumor chromosome.* Thus, any claim that one or the other event is more likely must be supported by other evidence. One could think of chromothripsis as another extreme scenario where *all* observed breaks are due to a single massive rearrangement. However, the statistical evidence from the rearranged genome cannot really distinguish chromothripsis from other sequences of copy neutral changes.
Inter-chromosomal rearrangements

Figure 3.10: Complex genome rearrangement event across multiple chromosomes. “abcdef” represents chromosome 5 and “ghij” represents chromosome 8. The gene fusion event involving DNA segments from chromosome 5 and 8 was discovered by McPherson et al. (2012).

While chromothripsis was originally claimed to occur on a single chromosome, the mechanism has been additionally claimed to affect loci clusters across multiple chromosomes Rausch et al. (2012); Molenaar et al. (2012); Kloosterman et al. (2011); Chiang et al. (2012a); Korbel and Campbell (2013). Genome rearrangement theory extends inversions to even apply to rearrangements affecting multiple chromosomes, and once again, a progressive sequence of ‘inversions’ (a more generalized version) can be used to explain these breakpoints as well. Once again, we are only claiming that the statistical evidence cannot distinguish between a multitude of copy neutral events, or a single, catastrophic event.

McPherson et al. discovered another gene fusion where ZDHHC11 and
RNF130 from chromosome 5 are fused together by a non-genic sequence from chromosome 8. In Figure S3.10, two extreme scenarios for rearrangement explain the breakpoints on the rearranged chromosomes. Here, chromosome 5 is labeled as “abcdef” and chromosome 8 is “ghij”. The ZDHHC11 and RNF130 genes are represented by segments ‘bc’ and ‘de’, respectively. The observed tumor chromosome 5 is “a(-c)(-b)hdef”. Again, one extreme is all the rearranged segments break simultaneously and form the rearranged tumor chromosomes (Figure S3.10). The other extreme is shown in Figure S3.10. First, suppose chromosome 5 and 8 are concatenated into “abcdefghij”. The observed tumor chromosome 5 can be created by an inversion of “bcdefg” (also known as reciprocal interchromosomal translocation), an inversion of “(-c)(-b)h”, and finally an inversion of “(-g)(-f)(-e)(-d)(-h)bc” (reciprocal interchromosomal translocation). Note that the term “inversions” generalizes in genome rearrangement theory and encapsulates the genomics term “interchromosomal translocation”. Also, the example demonstrates both scenarios equally explain the same rearranged chromosome.

What can and cannot be inferred from a complex rearranged genome

We are ready to tie together claims from the previous discussion.

1. **Available statistical evidence does not distinguish a single scenario of copy number neutral events (e.g., chromothripsis) as the source of genome rearrangement.**

2. **Explaining a rearranged genome using only inversions does not mean that inversions are the cause of the rearrangement. Instead, it suggests that an abundance of copy neutral events created the rearrangement.**

3. **Similarly, the number of inversion events is not representative of actual number of rearrangement events that occurred.**
4. *The copy neutral events are best generalized as k-break operations. In this scenario, inversions are 2-breaks, transpositions, and inverted transpositions are 3 breaks, and chromothripsis is the extreme example of an n-break. All combinations can generate complex rearranged genome, and deciding among those options needs additional evidence.*

Firstly, it is obvious that if there are two distinct and valid scenarios using the same number of inversions to explain the breakpoint data, then there is no way to discern which scenario rearranged the genome (Figure S3.11). In most cases, there are more than one scenario of inversions to explain breakpoint data. This observation is consistent with the theory that there is not necessarily a unique solution to the sorting of reversals problem Bafna and Pevzner (1996).

![Figure 3.11: Both scenarios of progressive inversions are equally likely. The breakpoints are the same as in Figure 3.9.](image-url)
Figures S3.8-3.10 demonstrate different scenarios explaining the same breakpoints may have fewer rearrangement steps. Indeed, most genome rearrangement problems are formulated to identify the minimum number of events Bafna and Pevzner (1996); Hannenhalli and Pevzner (1995); Alekseyev and Pevzner (2008); Kececioglu and Sankoff (1995). These formulations subtly assume the scenario that most “likely” occurred is the most parsimonious sequence of events. For species level genome comparison the parsimony assumption may be true. While it is tempting to assume that the parsimonious sequence of events is the more “likely” scenario for cancer rearrangements, it is inappropriate when allowing for complex rearrangement types.

In the above examples (Figures S3.8-3.10), each scenario was generated by permitting different rearrangement types. In genome rearrangement theory, the simplest rearrangement type is reversal (inversion) Hannenhalli and Pevzner (1995). Bafna and Pevzner Bafna and Pevzner (1996) developed an algorithm to find the minimum number of reversals to explain a rearranged genome (sorting by reversals). The same authors in a later work were able to solve for the minimum number of transpositions to explain a rearranged genome (sorting by transpositions) Bafna and Pevzner (1998). It is well established that transpositions are a more complex operation than reversals (see Figure S3.8) and less transpositions are necessary to explain a rearranged genome than reversals. There is little evidence that supports genomes evolved via only reversals or only transpositions. Thus, a scenario of reversals only is equally likely as transpositions only. In actuality, the true rearrangement process for genome evolution may have been a combination of both reversals and transpositions.

Likewise, the examples above propose two scenarios for rearranging a genome. The left scenario contains one extreme where a single event rearranges the genome
and the right scenario has the other extreme using many steps of inversions. Additionally, the breakpoint data can be explained by a combination of inversions and complex rearrangements (transposition).

Without setting a prior on the number of expected rearrangements, then it cannot be inferred which scenario is more likely to have occurred. Many more cancer genomes need to be accurately analyzed, before we can establish such priors.

**Claim:** Number of inversion events is not representative of actual number of rearrangement events that occurred.

This claim should now be obvious from the previous discussion. Alekseyev and Pevzner Alekseyev and Pevzner (2008) computed the bounds for number of events to rearrange a genome when limited to using only reversals and only transpositions. The authors formulate the rearrangement problem in terms of a breakpoint graph, where reversals are 2—break operations, transpositions are 3—break operations, and more complex rearrangements are \(k\)—break operations. They proved the minimum number of events necessary to explain breakpoints decreases drastically for larger \(k\). Since each scenario is an equally likely explanation and the number of events in each scenario varies greatly, we cannot use a single scenario, such as 2—breaks only, to be representative of the actual number of events that produced the rearranged genome.
3.10 Formal description of algorithms

**Stephens et al. simulation** Observed breakpoints are supplied as input to the simulation. In the process, breakpoints are added to a single chromosome in random order.

Each breakpoint is two positions with strandedness of the segments adjacent to the positions. $B$ is a set of breakpoints. Assume all segments are in the forward orientation and there is a parsimonious occurrence of rearrangements that create the observed breakpoints. Thus, $(+, -), (-, +), (-, -), (+, +)$ breakpoint strandedness correspond with deletion, tandem duplication, tail-to-tail inversion, and head-to-head inversion. Deletions and tandem duplications change the copy number state of the chromosomal segment between breakpoint positions (chromosome length changes). Inversions only inverted the segment and do not change the copy number state (no chromosome length change).

1: procedure StephenSimulateChrom($B$, breaks)
2: \hspace{1em} $c \leftarrow$ chromosome segmented by positions of $B$ and each segment is in the forward ($F$) direction.
3: \hspace{1em} while breaks $< b <$ breaks $+ 5$ do
4: \hspace{2em} rearr $\leftarrow$ random pop from $B$
5: \hspace{2em} $c \leftarrow c$ with rearr
6: \hspace{2em} $b \leftarrow$ unique or non-unique count of segment copy number changes.
7: \hspace{1em} end while
8: return $c, b$
9: end procedure

The simulation is repeated 35 times for copy number state changes of step(5, 85, 5).

**Kinsella et al. simulation** Chromosomes are simulated by starting with an interval and adding different rearrangements occurring at different rates with number of rearrangements and rearrangement sizes dictated by input distributions.
1: **procedure** KINSELLASIMULATECHROM(chromosome-length, rearr-num-distr, rearr-size-distr, inv-rate, del-rate, dup-rate, tandem-dup-rate, invert-dup-rate)
2: ▷ inv-rate, del-rate, dup-rate add up to 1.
3: ▷ tandem-dup-rate and invert-dup-rate are 0 − 1
4:  \( \text{breaks} \leftarrow \text{draw from rearr-num-distr} \)
5:  \( c \leftarrow [(1, \text{chromosome} - \text{length}),]. \)
6:  **for** \( i = 1 \ldots \text{breaks do} \)
7:    \( \text{rearr} \leftarrow \text{Inversion, Deletion, or Duplication randomly according to rearr} \)
8:    \( \text{rates and choose a random position.} \)
9:    \( \text{rearr} - \text{size} \leftarrow \text{draw from rearr-size-distr} \)
10:   \( \text{tandem} - \text{dup}, \text{invert} - \text{dup} \leftarrow \text{if Duplication randomly choose tandem} \)
11:   \( \text{or non-tandem and inverted or not inverted} \)
12:  \( c \leftarrow \text{splits 1 or 2 intervals in} \ c \text{based on} \ \text{rearr} \text{position and} \ \text{rearr} - \text{size}. \)
13:  \( c \leftarrow \text{delete, duplicate, and invert segments of} \ c \text{according the} \ \text{rearr} \)
14:  **end for**
15:  \( b \leftarrow \text{unique or non-unique count of segment copy number changes}. \)
16: **return** \( c, b \)
17: **end procedure**

**Kinsella et al. grimace**  Show an observed chromosome, supposed to have been shattered, can be created by progressive rearrangements. First, we convert a set of breakpoints into a signed permutation of blocks with respect to a reference genome (i.e. syntenic blocks). The signed permutation is then given to GRIMM to resolve a series of progressive rearrangements to reconstruct the reference genome.

For SNU-C1, breakpoints lie within a 80MB region (79749024 bp). Resulting blocks only cover a 40MB region (39652551 bp). No chromosome had two breakpoints with overlapping positions.
procedure PatelBreakpointsToPermutations(breakpoints)

▷ Partition chromosome by breakpoint positions

breakpoints ← {(pi, ei), (pj, ej) where pi, pj ∈ {1...L} and ei, ej ∈ {head, tail}}

V ← (1, head), (L, tail)

V ← V ∪ (p, tail), (p, head) ∀ p ∈ breakpoints

E ← breakpoints ∪ (pi−1, head), (pi, tail) 1 ≤ i ≤ 2n

G(V, E) ← incomplete breakpoint graph. Nodes have 1 or 2 edges.

Paths ← []

for G(B, P) ← ConnectedComponentsSubgraphs(G(V, E)) do ▷ B ⊂ V, P ⊂ E

if |P| == 1 then

P is a deleted segment in the rearranged chromosome

continue

end if

if only two nodes s, t ∈ B have exactly 1 neighbor in G then

▷ Subgraph is a path starting from s and ending at t

Paths ← DFS(G(V, E), s) ▷ list of edges

else

G(B, P) is a simple cycle ▷ tandem duplication

end if

end for

▷ Assemble complete path from (1, head) to (L, tail)

StartPath ← path has (1, head) ∈ Paths

EndPath ← path has (L, tail) ∈ Paths

Paths ← randomize ordering of Paths

▷ Every path starts with segment edge, alternates with breakpoint edges, and ends with a segment edge

SuperPath ← Chain(StartPath, Paths − {StartPath, EndPath}, EndPath)

L[e] ← label segment e ▷ order segment edges by lowest position and label 1, ... n

L[(pi+1, tail), (pi, head)] ← − L[(pi, head), (pi+1, tail)]

return L[e] for each segment e in SuperPath

end procedure
3.11 Acknowledgements

Chapter 3 is adapted from Kinsella M, Patel A, Bafna V. *The elusive evidence for chromothripsis*. *Nucleic Acids Research*. (2014). The dissertation author was a joint first author of this paper, and was responsible for computational research with regards to breakpoint analysis.
Chapter 4

InPhaDel: Integrative whole genome and proximity-ligation sequencing to phase deletions with variants

4.1 Introduction

Reference genomes are represented as a single set of chromosomes, but humans and many other organisms have diploid genomes. Most of the DNA sequence between homologous chromosomes is identical, with the exception of polymorphisms such as single nucleotide polymorphisms (SNPs) and deletions. More specifically, when diploid donor DNA is compared to a haploid genome reference, homozygous SNPs are genome positions where both donor DNA haploid copies differ from the reference and heterozygous SNPs are positions where only one of the copies differs. For example, a heterozygous site is a genome position
where the donor carries alleles A and C while the reference has only allele A. Similarly, a segment deleted in both copies of the donor DNA is homozygous, whereas a segment present in only a single copy is heterozygous. Next generation sequencing techniques (NGS) have enabled accurate genotyping of homozygous and heterozygous SNPs McKenna et al. (2010) and detection of deletions Mills et al. (2011). The problem of phasing heterozygous SNPs requires further analysis Bansal and Bafna (2008); Berger et al. (2014); Selvaraj et al. (2013). Phasing links alleles appearing at two heterozygous SNP sites to the same chromosome, and an example is given Figure 4.1. Phasing is useful for finding causes to diseases that only present in compound heterozygotes—that is, two mutant alleles need to appear on the same chromosome (cis) or different chromosomes (trans) for the disease to present.

Recently, isolating rare variants has been crucial for pinpointing the cause of rare diseases. In landmark studies, Ng et al. Ng et al. (2010) and Roach et al. Roach et al. (2010) found compound heterozygous single nucleotide mutations inactivate DHODH in cases of Miller syndrome. The mutations were identified by whole exome and whole genome sequencing of families with one or more children affected with Miller Syndrome. Sequencing a single affected alone is insufficient to whittle down the long list of putative causal variants. Additionally, two of the children in a quartet sequenced had compound heterozygous mutations in DNAH5, which explains why the children also presented with primary ciliary dyskinesia Roach et al. (2010). Inherited from the non-consanguineous parents, the deleterious mutations act in trans to inhibit gene function. In the families analyzed, each parent was heterozygous for a mutation and no unaffected siblings had compound heterozygous mutations.

Compound heterozygous mutations are not limited to inherited and single nucleotide mutations. Microarray analysis of large cohorts of schizophrenia and
autism spectrum disorder (ASD) individuals have shown 1721 megabases (Mb) de novo copy number variation (CNVs) to be a strong risk factor. In particular, loss of 22q11.2 has been shown to increase risk of schizophrenia. The hemizygosity combined with other rare mutations within the locus produces diverse phenotypes, including velo-cardio-facial syndrome and DiGeorge syndrome Stone et al. (2008). Similarly, compound heterozygous CNV and rare variants have been implicated in partial loss of function in a number of genes suspected to explain ASD Yu et al. (2013).

Hemizygosity may also involve small deletions. For example, bi-allelic mutations in \textit{ABCC6}, a 16.5 kilobase (kb) deletion of exons 23 – 29 compounded with more common mutations R1141X, R1164X, and R1138W, are known to cause psuedoxanthoma elasticum Ringpfeil et al. (2006). Since the single nucleotide mutations overlap the same deleted exons and additively depress \textit{ABCC6}, the gene was easily implicated in the disease. The compound heterozygous mutations that manifest these diseases have typically been mutations affecting biophysical properties of both disease-related genes copies. Long-range interactions, such as enhancers, may also affect expression of the same genes. Currently, these are difficult to implicate as the cause of a disease as any single rare disease sequencing study results in many putative combinations of long-range interactions and genes. Inferring whether the compound mutations act in \textit{cis} or \textit{trans} could greatly narrow down the list, and is only possible with long-range phasing.

Phasing of single nucleotide and deletion polymorphisms in an individual has been primarily accomplished by analyzing pedigrees Zhang et al. (2005) and population structure Delaneau et al. (2012). However, polymorphisms are mutations with allele frequency greater 1% in a population and rare variations (SNVs) with lower frequency will not be phased. Thus, phasing methods utilizing NGS
and unaware of population or parental genotypes are desired. A variety of computational techniques have been developed to phase SNPs directly from whole genome sequencing (WGS) Bansal and Bafna (2008); Halldorsson et al. (2004); Panconesi and Sozio (2004). More recently, a combination of proximity-ligation sequencing technique (HiC) Lieberman-Aiden et al. (2009) and HapCUT Bansal and Bafna (2008) has led chromosome long haplotypes Selvaraj et al. (2013). In WGS, paired-end sequencing reads are the ends of a DNA fragment and concordantly map to within 1kb on a reference. HapCUT is unable to achieve long haplotypes using only WGS reads because of the 1kb or less limitation of DNA fragment length. Thus, heterozygous SNPs spaced greater than 1kb apart are difficult to link. On the other hand, HiC experiments were designed to capture proximal DNA fragments. Paired-end reads from a HiC experiment are more complex with ends mapping to the same chromosome or different chromosomes Jin et al. (2013). A majority of the ends map to intrachromosomal positions < 2mb and are drawn from the same chromosome (cis). Thus, HiC provides greater than 1kb links needed for HapCUT to generate longer haplotypes.

While numerous SNV phasing methods have been developed for NGS, to our knowledge no one has developed a method to phase structural variations with SNVs without pedigree information. In this paper, we demonstrate phasing SNVs and deletions using only WGS and HiC sequencing experiments from a single diploid sample. As shown by Selvaraj et al. (2013), a scaffold of phased SNVs can be generated from WGS and HiC data. Also, a variety of algorithms have been developed to call deletions from WGS Mills et al. (2011). Therefore, we start with as input a list of heterozygous SNVs (genome positions), sequence of alleles from parent A alleles (pA), sequence of alleles from parent B (pB), deletion calls (pair of genome positions a,b), WGS reads, and HiC reads for each chromosome. Our
objective is to categorize deletion calls as belonging to $pA$, $pB$ or is homozygous. Additionally, calls inconsistent with the read data are unlikely to be deletions and fall into a separate category. We framed the problem as a multiclassification task and employed SVM and RandomForest supervised learning techniques to classify deletions.

4.2 Methods

Supervised learning of classes $C = \{\text{heterozygous } pA, \text{heterozygous } pB, \text{homozygous, incorrect}\}$ requires as input, feature vectors $f_1, f_2, \ldots, f_n$ and corresponding correct class labels $l_1, l_2, \ldots, l_n$ where $l_i \in C$. Once a model is learned, the model can then predict an arbitrary feature vector $f$ belongs to some class in $C$. The feature vector comprises of read counts from WGS and HiC datasets that distinguish deletion calls belonging to a single class in $C$ (see Figure 4.1).

4.2.1 Defining the feature vector

Firstly, WGS reads distinguish between heterozygous, homozygous, and incorrect deletions. Calling a segment $(a, b)$ deleted relies on 1) reads mapping discordantly where one end maps before $a$ and the second end maps after $b$, and 2) the segment $(a, b)$ having lower concordant read depth of coverage Medvedev et al. (2009). Therefore, we count the number of concordant reads mapping between $a$ and $b$ as a single feature (see Figure 4.1). To check for discordant read mappings, we count paired-end $+/−$ mapping reads between $1$kb segment pairs with starting chromosomal positions $(w − a, a), (w − b, b), \text{and } (w − a, b)$ where $w = 1$kb. In total, four features summarize the WGS data supporting a deletion call.

Secondly, we filter for WGS reads supporting the two haplotype. Reads
where one of the paired ends maps to an allele in pA are filtered into a WGS pA subset, and correspondingly reads overlapping pB go into a WGS pB subset. Similar to the full WGS read set, we count the number of reads between a and b, and paired-end reads mapping between 1kb segment pairs ($w - a, a$),($w - b, b$), and ($w - a, b$) for each WGS allele subset (see Figure 4.1). Thus, eight features summarize the WGS allele specific data supporting a deletion call belonging to pA or pB.

Thirdly, we filter for HiC reads supporting the two haplotypes into HiC pA and HiC pB subsets. For HiC, we count the number of paired-reads between a and b with the same orientation, $+/+$ or $−/−$, as a separate feature from the number of reads with opposite orientation (see Figure 4.1). Also, paired-end reads mapping with the same orientation between 1kb segment pairs with starting positions ($w - a, a$) and ($w - b, b$) are counted in separate features from paired-end reads mapping in opposite orientation. This accounts for twelve HiC allele specific features. Here, paired-end reads mapping between the 1kb segments pair with starting positions ($w - a, b$) is not included as a feature. Unlike WGS reads, HiC reads spanning around the deleted segment are not discordant and thus not informative of which allele a deletion appears.

Lastly, a majority of HiC reads fall within HindIII cut sites. To incorporate this additional signal, for the two nearest cutsites $s, t$ to deletion break $a$, where $s < a$ and $a < t < b$, the number of HiC pA subset reads mapping in the same orientation between the pair of 1kb segments starting at $s - w/2$ and $t - w/2$ is counted as a single feature. Similarly, reads counted for deletion break $b$, opposite orientation, and from HiC pB subset comprise features, which total eight features based on allele specific support from neighboring cut sites (see Figure 4.1). In total, there are twenty allele specific features derived from HiC data. All read counts are
normalized to reads per kilobase per million total reads mapped to chromosomes 19 and 20 (RPKM).

### 4.2.2 Training Dataset

Initially, we used 169 deletion polymorphisms annotated by HapMap 3 Consortium Altshuler et al. (2010) as a training dataset. The deletions were called in the European (CEU) trio, and the phasing of each deletion in the offspring NA12878 can be inferred from parents. In the HapMap project, the deletion polymorphisms are annotated by DNA segment copy number. If the copy number for a deletion polymorphism is 0 in the offspring and 0 or 1 in both the parents, then the deletion is labeled as homozygous. Whereas, deletion polymorphisms with copy number 1 in the offspring and 2 in one parent and 0 or 1 in the other parent are labeled as heterozygous. For a heterozygous deletion the matching $pA$ or $pB$ scaffold is inferred from the transmitted SNPs from the parents, also provided by HapMap. For the NA12878, there were in total 40 homozygous deletions and 79 heterozygous deletions. Lastly, incorrect deletions were found by selecting at random, 50 deletion polymorphisms where the copy number for each individual in the trio is 2. The models trained on the few HapMap examples performed poorly. (see Supplement Table 4.4). Simulation was necessary to boost the training dataset with true classifications and supporting read data. We simulated chromosomes with deletions, $(a_1, b_1), (a_2, b_2), \ldots, (a_{25}, b_{25})$ and simulated WGS and HiC reads sampled from the simulated chromosomes. We simulated ten chromosomes from reference chromosome 19 and ten chromosomes from reference chromosome 20. The location $a_i$ of the $i$th deletion was randomly selected from non-centromeric and non-telomeric regions and the size, $b_i - a_i$, is randomly drawn from the deletion length distribution from Mills et al. Mills et al. (2011).
For each created chromosome, we simulated WGS mapped reads to 36X depth of coverage using wgsim Li et al. (2009) and HiC mapped reads to 6X depth of coverage using our HiC read shuffler (see Supplement Section 4.8 and Figure 4.2). The total simulated read counts mapped for each chromosome is listed in Table 4.3. Since the simulated chromosome was created from only deletions, each position of the simulated chromosome aligns to a single position on the reference chromosome. Thus, the reference starting position for each simulated read is known. Simulated reads with a starting reference position \( p \) where \( a_i - 75 \leq p < a_i \) for some deletion \( i \) would form split-read mappings and are considered unmapped.

Classifications for deletions and their corresponding feature vectors are constructed by varying combinations of deletions \( D_i \) from chromosome \( i \) with read sets \( R_j \) for chromosome \( j \). For example, homozygous deletions are \( D_{11}, D_{12}, \ldots, D_{20} \) corresponding with feature vectors derived from \( R_{11}, R_{12}, \ldots, R_{20} \), respectively. In the simulated dataset, there are 250 homozygous deletions in the training dataset. Heterozygous deletions of class \( pA \) are \( D_1, D_3, D_5, \ldots, D_{19} \) corresponding with feature vectors from the union of each read set pair in \{ \( R_1, R_2 \), \( R_3, R_4 \), \ldots, \( R_{19}, R_{20} \) \}, respectively. For each read set pair, half the reads are randomly sampled to maintain 36X WGS read coverage and 6X HiC read coverage. Analogously, \( pB \) heterozygous deletions are \( D_2, D_4, \ldots, D_{20} \) with feature vectors derived from the same downsampled read set pairs as \( pA \). This generates 250 examples of feature vectors belonging to \( pA \) and 250 examples of \( pB \). Feature vectors labeled as incorrect deletions are generated by matching deletions calls \{ \( D_{19}, D_{20} \), \( D_{15}, D_{16} \), \( D_{11}, D_{12} \), \( D_{13}, D_{14} \), \( D_{17}, D_{18} \) \} with the above read set pairs, respectively. In total, there are 250 vectors in the incorrect class and 1000 simulated examples in total.

Out of the 1000 simulated examples, 205 had no reads supporting either \( pA \)
or \( pB \) and they were excluded from training. There were 41 HapMap examples with no reads supporting either \( pA \) or \( pB \), and these were excluded from training. This leaves 26 heterozygous \( pA \), 37 heterozygous \( pB \), and 17 homozygous examples with allele supporting reads in the HapMap dataset. Our training dataset is a combined simulated and HapMap dataset comprising of 240 heterozygous \( pA \), 248 heterozygous \( pB \), 159 homozygous, and 276 incorrect examples.

### 4.2.3 Training classifiers

We applied \( K \) Nearest Neighbors, SVM, and RandomForest techniques to our training dataset. For Nearest Neighbors, the brute-force algorithm was used and parameters \( K = 2, 4, 8, 16, 32 \) were tested. The parameter with the best accuracy in ten fold cross validation is used for classification. For SVM, a linear kernel SVM Vapnik (1995) is trained for each pair of classes, constituting a total of six models. The class prediction for a deletion is the class that received the most votes out of the six SVM models Knerr et al. (1990). In addition, the SVM regularization coefficients \( C = 1, 10, 100 \) were tested and ten-fold cross validation is used to select the best classifier. For RandomForest Breiman (2001), forests were trained with number of tree estimators 10, 20, 50, and 100, and tree max depth of 2, 5, 10, and 20. Again, the best parameters for the RandomForest classifier was selected using ten-fold cross validation. The scikit-learn python package Pedregosa et al. (2011) was used for \( K \) Nearest Neighbors, SVM, and RandomForest training, prediction, and model comparison.

### 4.2.4 Performance

The accuracy of a classifier is measured by the fraction of correctly classified predictions. In addition to a prediction, each classifier also reports the log probability
the deletion belongs to each class. The probability estimates can be used in a weighting strategy to alter the predictions that are made for each deletion. For a weighting strategy, the trade-off between correct and incorrect classifications can be assessed with a receiver operator curve (ROC). A standard ROC compares the true positive rate to the false positive rate for a binary classifier in a 2D plot. For multiple classes, we use a class reference ROC Fawcett (2004), which computes the trade-off of true classifications for a class with false classifications to any other class. To elaborate, suppose a classifier returns class \( c \) log probabilities \( s_1, s_2, \ldots, s_T \) for \( T \) examples. For the \( j \)th example the true class label is \( c(j) \in C \) and let \( t \) be a log probability threshold. The true positive rate for class \( c_i \) with threshold \( t \) is \( TPR(t) = \frac{\sum_{s_j \leq t} I(c(j) = c_i)}{P} \), where \( P \) is the number of examples with true class labels \( c_i \). Additionally, \( FPR(t) = \frac{\sum_{s_j \leq t} I(c(j) \neq c_i)}{T-P} \). The ROC curve can be plotted by varying \( t \) from \( s_1 \) to \( s_T \). Therefore, we can choose a particular weighting strategy that results in more correct classifications with respect to a particular class.

4.3 Results

4.3.1 Data

We demonstrate phasing on deletions from the European individual, NA12878. Deletions were called by HapMap 3 Altshuler et al. (2010) and Mills et al. Mills et al. (2011). We used the WGS data DePristo et al. (2011); Abecasis et al. (2012) and HiC data Selvaraj et al. (2013) generated from the same individual. Scaffolded phased SNPs were also provided by Selveraj et al. Selvaraj et al. (2013).

The simulated WGS read data was generated by wgsim Li et al. (2009) and simulated HiC read data was created by our custom HiC read shuffler (see Supplemental Section 4.8). HiC experiments create a highly distinct pattern of
paired-end reads compared to WGS Jin et al. (2013). In WGS, paired-end reads are expected to map concordantly with the lower position end mapping to the + strand, and the higher position end mapping to the − strand. Contrastingly, all four paired end read combinations (+/−, −/+ , −/− , +/+ ) are concordant in HiC experiment. Typically, +/− and −/+ read orientations are dominant when the ends map to less than 25kb, which likely represent intrachromosomal interactions Jin et al. (2013). Our HiC simulator produces a similar distribution of paired-end orientations at each distance as the reads generated from a real HiC experiment, as shown in Figures 4.2-4.3. In addition, the simulator accounts for a majority of the paired-end reads having ends mapping within HindIII cutsites. The cutsite bias results in non-uniform read coverage at greater than 25kb distances in Figure 4.2. For our HiC simulator, we chose not to simulate chromosomal spatial interactions as the effects are only observed at distances greater than 25kb. At a distance greater than 25kb on chromosome 20 there were only 2 million reads. This averages to 10.3 reads per 100kb square bin (see Figure 4.2). More importantly, most of the reads lie at distances less than 40kb, and the HiC shuffler and real HiC experiments appear identical at these distances (see Figure 4.2).

4.3.2 Accuracy in simulated dataset

We first tried Nearest Neighbors classification restricted to 6 features. The 6 features summarize read counts supporting or negating homozygous, incorrect, pA or pB classes (see Supplement Section 4.9.1). The classifier did not perform well on the Hapmap 3 dataset. Even Nearest Neighbors classification using the full 32 features had markedly lower performance compared to both SVM and RandomForest methods (see Figure 4.4). Of the 923 combined training deletions, both SVM and RandomForest performed similarly well using all 32 features with
97.2% and 99.6% classification accuracy, respectively. As shown in Figure 4.4, the few misclassification errors in RandomForest occurred in classifying heterozygous examples. For SVM, the misclassification errors appear across all classes.

In addition, we hypothesized allele specific features from WGS reads are insufficient for accurate classification. We trained models using three feature subsets: WGS unfiltered data, WGS unfiltered plus WGS pA and pB filtered data, and unfiltered WGS plus HiC pA and pB filtered data. Respectively, the subsets had 4, 12, and 24 features (see Figure 4.1). The SVM and RandomForest models trained on only WGS unfiltered data was a control with no allele specific features. The control models could only classify homozygous and incorrect classes accurately and had the expected lower 73.1% and 72.2% accuracies, respectively. The remaining SVM and RandomForest models trained on allele specific feature subsets had similar high classification accuracy (see Figure 4.4). The RandomForest model built on WGS plus HiC data had no classification errors, where as the SVM model on the same feature subset had 95.7% accuracy. The WGS and HiC allele specific feature subsets differed in the number of deletions with no allele supporting reads. The HiC feature subset had 22.2% more deletions than the WGS feature subset. Of all the correct predictions between the RandomForest HiC and WGS feature subset models, 23.4% were phased only by HiC compared to 6.1% by WGS (see Figure 4.4f).

Also, we tested if increasing HiC read coverage to 12X or WGS read coverage to 76X would impact the number of examples that could be phased. We found neither increase had a significant effect on accuracy. Surprisingly, the number of simulated deletions with no supporting reads did not change (see Table 4.1). Doubling the WGS coverage did not gain more reads supporting the phasing of the deletion. Doubling the HiC read coverage gained 4.2% more deletions with
supporting HiC reads, however these deletions were already supported by WGS reads and overall the gain was 0.7%.

Lastly, our method depends on accurate SNP phasing and SV calls. We tested the tolerance of the trained RandomForest model to increasing errors in deletion breakpoints \(a, b\). Starting with the true \(pA\) and \(pB\) deletion call set, we modified \(a\) in decrements of 100bp and \(b\) in increments of 100bp for each deletion to create erroneous call sets. Here, deletions with no allele specific supporting reads were recorded as misclassifications. This allows for a fair comparison between calls made from different deletion error sets. The RandomForest model retained high accuracy when the errors for \(a\) and \(b\) were less than 200bp (see Figure 4.6). Additionally, across each category of deletion sizes, the trained model had the largest loss of accuracy when \(a\) and \(b\) errors were both greater than 300bp.

4.3.3 Accuracy in deletions from NA12878

Our method performs well on the 169 deletion polymorphisms from HapMap 3 dataset Altshuler et al. (2010). On the HapMap 3 dataset, RandomForest and SVM had 100% and 89% accuracy, respectively. In 10 of the 13 inconsistent classifications between SVM and the HapMap, SVM predicted there was no evidence for a deletion in our data set (see Supplement Table 4.5). Indeed, upon manual inspection of the WGS and HiC reads, SVM correctly attributed a strong incorrect deletion score to 7 of the above deletion polymorphisms (see Supplement Figure 4.12. Despite these observed inconsistencies, RandomForest yields predictions consistent with the HapMap 3 deletion classifications, possibly due to unavoidable overfitting of the RandomForest model to the HapMap 3 dataset. Outside of these few inconsistencies, SVM and RandomForest generally provide consistent deletion predictions.

In a recent study demonstrating single cell sequencing in the NA12878 line Fan et al.
(2011), 15 of the HapMap 3 heterozygous deletions were independently validated by PCR. The SVM and RandomForest methods correctly predicted 13 of these deletions. The remaining 2 deletions had no WGS or HiC reads supporting a particular allele, and no prediction was made.

As an independent validation, we analyzed our predictions for the July 2010 deletion set released by the 1000 Genomes Project Abecasis et al. (2012); Mills et al. (2011). The release set consists of merged calls curated from diverse computational methods that called deletions on sequencing data Mills et al. (2011). 153 out of the 1170 merged calls with size greater than 1kb could be phased to $pA$ and $pB$ by assigning the deletion as present or absent in each individual in the European trio (see Supplement Figure 4.10). If any computational approach found a deletion in the mother, father, or child matching a merged call, then the merged call is said to be present in the corresponding individual. Only merged calls that are present in two of the three individuals can be phased. Merged calls that are present in all three individuals can be either homozygous or heterozygous in the child, thus these are not considered in validation (see Supplement Figure 4.10 for details). RandomForest method accurately predicted 74.8% of the phased deletions variants (see Figure 4.7 and Table 4.2). To distinguish between possible errors in the Mills et al. Mills et al. (2011) deletion (Mills) phasings and our predictions. We measured accuracy on Mills deletions that overlapped with previously annotated gold standard deletions Mills et al. (2011) (Mills-gs). The accuracy of the RandomForest model increased to 91.1% in the Mills-gs dataset. Additionally, to find errors strictly associated with the machine learning technique, we removed Mills deletions where both classifiers made the same prediction that was inconsistent with Mills (Mills-confident). Expectedly, the RandomForest accuracy increased to 95.3%.

Furthermore, we use class reference ROC Fawcett (2004) to test the perfor-
mance of RandomForest and SVM probability estimates for deletion classes $pA$ and $pB$ (see Figure 4.8). In addition to having a better accuracy, RandomForest outperformed SVM when scoring predictions on the ROCs. We tried both log probability and delta score for determining ROC cut-offs and found they perform similarly for each classifier. Delta score is defined as the difference between the two best class log probabilities for a given deletion. The log probability and delta score performed similarly because they are correlated metrics. Using a scoring threshold, the ROC shows a higher accuracy can be achieved with respect to a single class.

Since there were few misclassification errors in Mills Mills et al. (2011), we inspected reads falling within regions of the inconsistencies using Savant Genome Browser Fiume et al. (2010). The inconsistencies fell largely into three categories. Firstly, there were 18 inconsistencies where there was little evidence to support a deletion occurred in our WGS and HiC datasets (see Supplement Figure 4.13). Secondly, 2 out of the 5 inconsistencies called homozygous had strong support by our read datasets (see Supplement Figure 4.14). Lastly, 15 inconsistencies were due to insufficient WGS and HiC read support (see Supplement Figure 4.15). More conservative scoring thresholds can be used with SVM and RandomForest to eliminate the low read evidence false classifications. However, as shown with the ROC, setting a scoring threshold will sacrifice numerous correct classifications as well.

4.3.4 Putative impact to allele-specific expression sites

Clearly, deletions interrupting a gene’s transcriptional sequence must diminish the gene’s expression. For a maternally transmitted deletion, the functional impact must be to favor expression of the paternal gene copy, including the paternal alleles in the gene. Similarly, a deletion overlapping a transcription factor binding
site could force transcription factors to bind to the homologous chromosome’s binding site. Rozowsky et al. (2011) compiled a list of genes under allele-specific expression by analyzing RNA-seq experiments mapped to a diploid NA12878 genome. Of the known genes on autosomal chromosomes, 11 overlapped with deletions in the Mills et al. (2011) call set. However, only 2 gene and deletion pairs had allele-specific expression consistent with our prediction of deletion phasing. In comparison, 6 gene and deletion pairs had allele-specific expression consistent with the Mills et al. (2011) phasings. Upon closer inspection, 5/11 deletions mapped to the introns of a single gene, SKA3 (see Figure 4.9). These deletions had no overlap with the annotated gold standard in Mills et al. (2011), but were identified by multiple computational approaches using the Mills et al. 1000 Genomes Project data. The RandomForest method predicted 3 of the deletions as incorrect and the SVM method predicted all 5 as incorrect. The underlying WGS and HiC data support these predictions that no deletion is present in the gene (see Supplement Figure 4.13). There are two logical explanations for this inconsistency. There whole genome sequencing sample may be contaminated with SKA3 cDNA fragments and reads generated from the transcript were reported by the computational methods as true deletions. Alternatively, mismapping of reads between SKA3 exons can cause SV calling algorithms to incorrectly predict deletions. However, the exons have low similarity and good mappability, and mismapping reads is an unlikely hypothesis.

In general, both Rozowsky et al. and Mills et al. have exceptional accuracy and 75% of our predictions are consistent with Mills et al. release set. Although, only a few inconsistencies can lead to a strong impact on downstream analysis—SKA3 being the only example found in an otherwise well curated datasets. Rozowsky et al. used the Mills et al. phased variant calls to create a diploid NA12878 genome.
RNA-seq and RNA polymerase II ChIP-seq data was then analyzed using the diploid reference to call allele-specific expression and binding. This explains the consistency between Mills et al. deletion calls as paternally transmitted and the Rozowsky et al. maternal homolog specific SKA3 expression and RNA polymerase II binding adjacent to SKA3.

With regards to allele specific binding, there was only one deletion, chromosome 17:38,793,571 − 38,795,791 that overlapped with 14 SNPs annotated with allele-specific cMyc, RNA Polymerase II, and NFκ B binding. The allele-specific binding is consistent with the phasing of the deletion predicted by SVM and RandomForest methods, and annotated by Mills et al. (2011).

4.4 Predictions for all NA12878 deletions.

We also predicted deletion classifications for HapMap 3 deletions that were somatic in NA12878. In total, we categorized 70 heterozygous, 17 homozygous, and 3 incorrect deletions in the dataset, using RandomForest. There were 42 deletions with no allele supporting read data and calls were made only in 68% of HapMap 3 deletions (see Supplement Data). SVM corroborated 80% of the RandomForest predictions. We also made predictions for any deletion found in the NA12878 individual by Mills et al. Mills et al. (2011). The predictions included all cases where a deletion was found in NA12878 and her parents. Previously, these deletions were not analyzed as they could not be phased by Mills et al. data alone. We found 276 heterozygous, 57 homozygous, and 151 incorrect deletions in the dataset. Previously, there were 185 deletions with no allele specific read support and calls were only made in 72.4% of the Mills et al. data. Again, SVM corroborated 80% of the RandomForest predictions.
In our simulations, 85% of the deletions were supported by allele specific reads 4.1. The simulations do not explicitly model for read mismappings, therefore a smaller fraction of deletions in the empirical datasets is expected.

4.5 Discussion

Others Bansal and Bafna (2008); Selvaraj et al. (2013); Mills et al. (2011) have developed methods for calling deletions and phasing SNPs from WGS and HiC experiments performed on a single individual, in an effort to understand variation in the diploid human genome. Continuing in this endeavor, our method is able to assign deletions to a phased SNP scaffold using only WGS and HiC reads from the same single individual. Medically, phasing deletions with SNPs are important for identifying causal genes implicated in rare diseases Roach et al. (2010); Ng et al. (2010) and neurological disorders Stone et al. (2008); Yu et al. (2013). Numerous CNVs have been associated with increased risk for schizophrenia and ASD. These disorders and their observed diverse phenotypes could be explained by compound heterozygous CNVs and rare variants.

Our approach addresses deletion phasing to confirm whether the heterozygous deletion acts in cis or trans with other heterozygous variants. We formulate the task as a multiclassification problem where deletions can be phased to either chromosome (pA or pB), homozygous, or unsupported by the read data. We used SVM and RandomForest approaches to solve this problem and found RandomForest provided superior prediction scoring and classification accuracy. We demonstrated accuracy on the NA12878 European individual, who was part of a trio analyzed by microarray experiments (HapMap 3 Consortium Altshuler et al. (2010)) and whole genome sequencing (1000 Genomes Project Mills et al. (2011)). The data on trios
provides a validation set of deletions with known phasings or known homozygosity. RandomForest achieves 100% accuracy on HapMap 3 deletion polymorphisms and 91% accuracy on Mills et al. (2011) deletion variants common with previously known deletions. In addition, we found the inconsistencies between RandomForest calls and the deletion variant release set can be explained in some cases by low read support. In other cases, the read data was inconsistent with the Mills et al. deletion call. As shown in our results of the SKA3 gene, a few errors in the release dataset could lead to potential errors in downstream analysis of allele-specific expression and binding.

To train robust SVM and RandomForest models required simulation of WGS and HiC reads sets on chromosomes with known deletions. To our knowledge, there is no known HiC read simulator. We developed a method for simulating HiC reads through a computationally intensive read shuffling approach (see Supplement Section 4.8). The simulations not only improved robustness of the supervised learning models, but also facilitated additional analysis of capabilities of HiC and WGS experiments. Using simulations, we show that RandomForest tolerates breakpoint errors up to 400bp and then accuracy deteriorates. Notably, deletions larger than 10kb can still achieve 70% classification accuracy with breakpoint errors greater than 400bp. More importantly, simulations show HiC read data contributes to uniquely phase 23% of deletions compared to 6% by WGS data. HiC enables distant linkages between distant mutations, in our case SNPs and deletions. The same concept has been previously applied to phase distant SNPs Selvaraj et al. (2013).

The focus of this paper has been on deletions, which have primarily been the structural variations analyzed by next generation sequencing. The next step is to phase duplications. Our multiclassification problem formulation, which works
well for deletions is not ideal for duplications. Duplications encompass multiple copy number states and may or may not occur tandemly. Thus, there are numerous classes for a duplication event. To handle the additional classes requires more data and introduces new kinds of errors. For example, a region duplicated to a homologous chromosome should be phased to both chromosomes $pA$ and $pB$. Further investigation is needed to address phasing of structural variation and other computational methods may prove to be more valuable, however it is clear that distant interactions provided by HiC will be necessary.

4.6 Tables

Table 4.1: Increasing read coverage does not significantly improve coverage of deletions for phasing. The table shows the number of simulated $pA$ and $pB$ deletions with no supporting reads for each HiC and WGS coverage, set and feature subset. The total number of deletions is 500.

<table>
<thead>
<tr>
<th>WGS cov.</th>
<th>HiC cov.</th>
<th>All</th>
<th>WGS $pA$, $pB$</th>
<th>HiC All</th>
<th>HiC $pA$, $pB$</th>
<th>HiC cutsite $pA$, $pB$</th>
</tr>
</thead>
<tbody>
<tr>
<td>36x</td>
<td>6x</td>
<td>75</td>
<td>163</td>
<td>93</td>
<td>93</td>
<td>467</td>
</tr>
<tr>
<td>72x</td>
<td>6x</td>
<td>77</td>
<td>165</td>
<td>93</td>
<td>93</td>
<td>467</td>
</tr>
<tr>
<td>36x</td>
<td>12x</td>
<td>72</td>
<td>163</td>
<td>76</td>
<td>76</td>
<td>458</td>
</tr>
</tbody>
</table>
Table 4.2: RandomForest had fewer misclassification errors than SVM. Both methods had increased accuracy when measuring accuracy on Mills-gs compared to Mills, even though both are independent test sets. The table shows confusion matrices for SVM and RandomForest methods on Mills et al. deletions and filtered Mills deletions, Mills-gs and Mills-confident. Parenthesis contain percentages.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mills-confident</th>
<th>Mills-gs</th>
<th>Mills</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hom. incorrect</td>
<td>pA</td>
<td>pB</td>
</tr>
<tr>
<td>RandomForest</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pA</td>
<td>0 (0.00)</td>
<td>45 (0.96)</td>
<td>2 (0.04)</td>
</tr>
<tr>
<td>pB</td>
<td>1 (0.02)</td>
<td>2 (0.03)</td>
<td>56 (0.95)</td>
</tr>
<tr>
<td>Total</td>
<td>1 (0.01)</td>
<td>47 (0.44)</td>
<td>58 (0.55)</td>
</tr>
<tr>
<td>SVM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pA</td>
<td>0 (0.00)</td>
<td>3 (0.06)</td>
<td>6 (0.13)</td>
</tr>
<tr>
<td>pB</td>
<td>1 (0.02)</td>
<td>0 (0.00)</td>
<td>51 (0.86)</td>
</tr>
<tr>
<td>Total</td>
<td>1 (0.01)</td>
<td>38 (0.36)</td>
<td>57 (0.54)</td>
</tr>
<tr>
<td>SVM Mills-confident</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pA</td>
<td>1 (0.02)</td>
<td>33 (0.80)</td>
<td>5 (0.12)</td>
</tr>
<tr>
<td>pB</td>
<td>2 (0.05)</td>
<td>0 (0.00)</td>
<td>36 (0.95)</td>
</tr>
<tr>
<td>Total</td>
<td>3 (0.04)</td>
<td>33 (0.42)</td>
<td>41 (0.52)</td>
</tr>
<tr>
<td>SVM Mills-gs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pA</td>
<td>3 (0.05)</td>
<td>38 (0.60)</td>
<td>9 (0.14)</td>
</tr>
<tr>
<td>pB</td>
<td>2 (0.03)</td>
<td>19 (0.26)</td>
<td>51 (0.71)</td>
</tr>
<tr>
<td>Total</td>
<td>5 (0.04)</td>
<td>38 (0.28)</td>
<td>60 (0.44)</td>
</tr>
</tbody>
</table>
4.7 Figures

![Phasing example and description of feature vector](image)

**Figure 4.1:** Phasing is necessary to identify heterozygous mutations are in *cis* or *trans*. In (a) there are eight possible diploid genome interpretations given only genotype inputs. In (b) WGS and HiC reads in 1kb intervals around a putative deletion are used as features. The read counts are made from subsets of read data: WGS, WGS \( pA \), WGS \( pB \), HiC \( pA \), HiC \( pB \), HiC cutsites \( pA \), HiC cutsites \( pB \). Coloring of the interval indicates the orientation of paired-end reads that are counted. Read counts are normalized by interval length and number of million reads (RPKM).
Figure 4.2: The simulated HiC reads have the same distribution of counts as the real HiC dataset as shown by the visualization of counts across 62mb length chromosome 20 with 100kb by 100kb bins. The x-axis marks the binned position of the left-most mapped end of the read and the y-axis marks the binned position of the right-most mapped end of the read. Above the diagonal are read counts for paired-end orientations, ++, -- (green), --+ (yellow), and +-- (blue). Below the diagonal, is the total read count (red). The top row shows real HiC data from chromosome 20 Selvaraj et al. (2013). The bottom row shows similar read counts, but was generated by shuffling the HiC reads mapping to chromosome 20.
Figure 4.3: Within 40kb windows, the simulation and real dataset have nearly indistinguishable read count distributions, notably preserving counts of each orientation type as a function of genome distance between paired-ends Jin et al. (2013). The close-up is a sum of read counts over 100 windows with 200bp by 200bp binning. The x-axis marks the binned position of the left-most mapped end of the read and the y-axis marks the binned position of the right-most mapped end of the read. Above the diagonal are read counts for paired-end orientations, \(+/-, -/+\) (green), \(-/-, +/+\) (yellow), and \(+/-\) (blue). Below the diagonal, is the total read count (red). The top row shows real HiC data from chromosome 20 Selvaraj et al. (2013). The bottom row shows the same pattern, but was generated by shuffling HiC reads mapping to chromosome 20.
**Figure 4.4**: Simple classifiers are insufficient for accurate phasing. a-c) Accuracy of Nearest Neighbors with 6 features, 32 features, and RandomForest with 32 features on the HapMap 3 dataset. The classes are $p_A$, $p_B$, homozygous (hom.) and incorrect. The width of each column is proportional to the number of examples in each class. d) Accuracy of SVM models trained on 32, 12, and 24 features shown as rows. e) Accuracy of RandomForest models.

**Figure 4.5**: HiC reads contribute to correctly predict 23% of the deletion classes in simulation. The contribution of HiC reads increases to $\approx 40\%$ for deletions in HapMap 3 (g) and Mills (h). Breakdown of the correct predictions made by only RandomForest HiC feature subset model (red), WGS feature subset model (blue), or both (magenta).
Figure 4.6: RandomForest model retains accuracy with small breakpoint error in the deletion calls (a, b) and is particularly tolerant to errors for 10kb or greater deletions. To test error, a is decreased from 0bp to 700bp and b is increased from 0bp to 700bp in increments of 100bp. Here, deletions with no supporting reads are counted as misclassifications. a) Accuracy of 144 pA and pB deletions of size 1 – 5kb. b) Accuracy of 56 pA and pB deletions of size 5 – 10kb. c) Accuracy of 50 pA and pB deletions greater than 10kb.

Figure 4.7: RandomForest has good accuracy on the Mills et al. deletion variants. a) Accuracy of RandomForest classifier with 32 features on the Mills et al. Mills et al. (2011) merged deletions on NA12878 (Mills). b) Accuracy of RandomForest classifier after filtering Mills et al. with previously identified gold standard deletions Mills et al. (2011) (Mills-gs).
Figure 4.8: RandomForest provides better scores than SVM as shown by the greater area under the curve (AUC). The ROCs are shown across the three Mills deletion sets: the Mills et al. deletion release set (Mills), gold standard filtered set (Mills-gs), and read data inconsistency excluded set (Mills-confident).
Figure 4.9: RandomForest and SVM predicted the suspicious deletions in the \textit{SKA3} gene as incorrect. The read data also did not strongly corroborate numerous deletions (see Supplement Figure 4.13. Genome browser image of \textit{SKA3} with allele-specific expression and binding SNPs from Rozowsky et al. Rozowsky et al. (2011). Deletions from Mills et al. Mills et al. (2011) are annotated on a separate track.
4.8 HiC Shuffling Simulator

The HiC read simulator was engineered to (1) preserve the distribution of read orientations at distances partitioned by bin edges $l_0, l_1, \ldots, l_n$, and (2) preserve the number of reads within $w$ of a restriction enzyme cutsite. Therefore, we took a read shuffling approach. Each HiC paired end read $(x, y) \in R$ mapped to a sequence $S$ is shuffled to a new location in sequence $S'$ under the constraints given above. For simplicity, let $S$ and $S'$ have the same length $L$ and positions for cutsites $H$.

Algorithm 2 for shuffling reads has a runtime in $O(|R| \log |R|)$. However, this requires holding the $L \times L$ matrices $Z$ and $Index$ in main memory. For chromosome 20, $L \approx 6 \times 10^7$ and thus requires 360 terabytes of main memory ($3.6 \times 10^{15}$ bytes).

Therefore, we applied the following three optimizations to create a feasible HiC read shuffler. Modification 3 is an approximation, but does not strongly affect the resulting simulated reads as observed in Figure 4.2.

1. Since $R$ is sorted such that $y_{i-1} - x_{i-1} < y_i - x_i$ for the $i$th read. Reads are processed per $b$th bin. For the $b$th bin, only $Z[x, y]$ for $l_{b-1} \leq y - x < l_b$ and $Index(Z, b, z)$ for $z \in \{0, 1, 2\}$ need to be loaded into memory.

2. Positions in $R, H, H'$, and $l_0, \ldots, l_n$, are transformed by $\lfloor \frac{p}{100} \rfloor$ and then for each read $R'$ we add $Random(1 \ldots 99)$.

3. For bins greater than $b'$ we only shuffle mapped read with both ends lying in cutsites. At greater distances, HiC reads tend to be sparser and the frequency of reads at mapped positions in zero or one cutsites is negligible. This is acheived using a cutsite only data structure as demonstrated in Algorithm 3.
Algorithm 2 HiCShuffling algorithm

1: \textbf{procedure} HiCShuffle(HiC reads $R$, cutsites $H, H'$, sequences $S, S'$, bin edges $B$)
2: \hspace{1em} $\triangleright$ Randomly shuffles reads $R$ on $S$ to new locations $R'$ on $S'$
3: \hspace{1em} $Z_S \leftarrow \text{ConstructZ}(w, H, |S|)$ \hspace{1em} $\triangleright$ Z matrix with cutsite counts for $[x, y]$
4: \hspace{1em} $Z_{S'} \leftarrow \text{ConstructZ}(w, H', |S'|)$
5: \hspace{1em} Let $\text{Index}(Z, b, z)$ be the $(x, y)$ coordinates where $l_{b-1} \leq y - x < l_b$ and $Z[x, y] = z$
6: \hspace{1em} $b = 1$
7: \hspace{1em} \textbf{for} $(x, y) \leftarrow \text{Sorted}(R)$ \textbf{do} \hspace{1em} $\triangleright$ sort $R$ by ascending $y - x$
8: \hspace{1em} \hspace{1em} $b + +$ if $l_b \leq y - x$
9: \hspace{1em} \hspace{1em} $z \leftarrow Z_S[x, y]$ \hspace{1em} $\triangleright$ get number of cutsites for read
10: \hspace{1em} \hspace{1em} $R' \leftarrow \text{Random}(\text{Index}(Z_{S'}, b, z))$ \hspace{1em} $\triangleright$ choose coord from list at random
11: \hspace{1em} \textbf{end for}
12: \hspace{1em} \textbf{return} $R'$
13: \textbf{end procedure}
14: \textbf{procedure} ConstructZ($w, H, L$)
15: \hspace{1em} $Z \leftarrow L$-by-$L$ zero matrix
16: \hspace{1em} \textbf{for} $h \leftarrow H$ \textbf{do}
17: \hspace{1em} \hspace{1em} $Z[w - h \ldots h + w, w - h \ldots L] + = 1$ \hspace{1em} $\triangleright$ All positions $x$ that are in cutsite $h$
18: \hspace{1em} \hspace{1em} $Z[w - h, \ldots L, w - h \ldots h + w] + = 1$ \hspace{1em} $\triangleright$ All positions $y$ that are in cutsite $h$
19: \hspace{1em} \textbf{end for}
20: \hspace{1em} \textbf{return} $Z$ a matrix that gives the count $\{0, 1, 2\}$ of cutsites for each read coordinate.
21: \textbf{end procedure}
Algorithm 3 HiCShuffling algorithm for reads at cutsites and bins greater than $b'$

1: \textbf{procedure} \textsc{HiCShuffleCutsOnly}(HiC reads $R$, cutsites $H, H'$, sequences $S, S'$, threshold $b'$)
2: \hspace{1em} $\triangleright$ Randomly shuffles reads $R$ on $S$ to new locations $R'$ on $S'$
3: \hspace{1em} Let $D_{b,s}(s,t)$ be the $(x, y)$ coordinates where $h_s - w \leq x < h_s + w$, $h_t - w \leq y < h_t + w$, and $l_{b-1} \leq y - x < l_b$
4: \hspace{1em} $\triangleright$ For each $b$, $D$ is a list of lists sorted by $s$ and then $t$, finally pointing to a binary matrix.
5: \hspace{1em} $b = b'$
6: \hspace{1em} \textbf{for} $(x, y) \leftarrow \text{Sorted}(R)$ \textbf{do}
7: \hspace{2em} $b \leftarrow b + 1$ if $l_b \leq y - x$
8: \hspace{2em} $\triangleright$ Check first if $(x, y)$ falls in a cutsite on $S$
9: \hspace{2em} $i \leftarrow \text{BinarySearch}(H, x)$
10: \hspace{2em} $j \leftarrow \text{BinarySearch}(D_{b,s}(i,), y)$
11: \hspace{2em} \textbf{if} $(x, y) \in D_{b,s}(i,j)$ \textbf{then}
12: \hspace{3em} $R' \leftarrow \text{Random}(\sum_{s,t} D_{b,s'}(s,t))$ \hspace{1em} $\triangleright$ choose coord from concatenated lists at random
13: \hspace{2em} \textbf{else}
14: \hspace{3em} Continue
15: \hspace{2em} \textbf{end if}
16: \hspace{2em} \textbf{end for}
17: \hspace{1em} \textbf{return} $R'$
18: \textbf{end procedure}
4.9 InPhaDel supplement

4.9.1 Simplifying extensive 32 features to 6

Initially, we attempted the simplest model with 6 features. WGS data can distinguish between homozygous, heterozygous, and incorrect deletions. Specifically for a deletion $(a, b)$, the incorrect, heterozygous, and homozygous call depends on the number of concordant WGS reads in $(a, b)$ (feature 1 in Figure 4.1) matching either average read depth coverage, half the average, or no read depth coverage, respectively. Conversely, only observing discordant reads around the interval $(a, b)$ (feature 2 in Figure 4.1) would support a homozygous call, and negate incorrect or heterozygous calls.

The remaining four features are simple summations of reads supporting or contradicting deletions belonging to $pA$ or $pB$. For a deletion $(a, b)$, the sum of WGS and HiC reads in $(a, b)$ (features 5, 13, 14, 19, 20) and reads around each break $a$ and $b$ (features 7, 8, 15 – 18, 25 – 28) would support $pA$. Whereas, the number of WGS discordant reads around $(a, b)$ (features 6) would contradict the deletion belonging to $pA$. Similarly, two summation features are constructed for supporting or contradicting a deletion belonging to $pB$. 
### 4.9.2 Tables

**Table 4.3:** Mapped read counts from simulated chromosomes.

<table>
<thead>
<tr>
<th>chromosome</th>
<th>WGS</th>
<th>HiC</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr19 1</td>
<td>22998062</td>
<td>3021768</td>
</tr>
<tr>
<td>chr19 2</td>
<td>22997526</td>
<td>3021768</td>
</tr>
<tr>
<td>chr19 3</td>
<td>23001674</td>
<td>3021766</td>
</tr>
<tr>
<td>chr19 4</td>
<td>22999354</td>
<td>3021765</td>
</tr>
<tr>
<td>chr19 5</td>
<td>22998892</td>
<td>3021761</td>
</tr>
<tr>
<td>chr19 6</td>
<td>23003694</td>
<td>3021764</td>
</tr>
<tr>
<td>chr19 7</td>
<td>22999709</td>
<td>3021767</td>
</tr>
<tr>
<td>chr19 8</td>
<td>23001677</td>
<td>3021765</td>
</tr>
<tr>
<td>chr19 9</td>
<td>23002088</td>
<td>3021752</td>
</tr>
<tr>
<td>chr19 10</td>
<td>22999500</td>
<td>3021762</td>
</tr>
<tr>
<td>chr20 11</td>
<td>23002458</td>
<td>4184686</td>
</tr>
<tr>
<td>chr20 12</td>
<td>22995638</td>
<td>4193912</td>
</tr>
<tr>
<td>chr20 13</td>
<td>22996220</td>
<td>4189989</td>
</tr>
<tr>
<td>chr20 14</td>
<td>23000178</td>
<td>4196526</td>
</tr>
<tr>
<td>chr20 15</td>
<td>22994652</td>
<td>4191554</td>
</tr>
<tr>
<td>chr20 16</td>
<td>23001578</td>
<td>4194441</td>
</tr>
<tr>
<td>chr20 17</td>
<td>22994329</td>
<td>4194457</td>
</tr>
<tr>
<td>chr20 18</td>
<td>22996746</td>
<td>4187319</td>
</tr>
<tr>
<td>chr20 19</td>
<td>22996853</td>
<td>4185965</td>
</tr>
<tr>
<td>chr20 20</td>
<td>23002791</td>
<td>4190726</td>
</tr>
</tbody>
</table>
Table 4.4: Testing classifier on a dataset that was not used in training results in poor classification accuracy. Table shows counts of examples where rows are classes, homozygous (hom.), incorrect (inc.), parent A (pA), and parent B (pB), and columns are predicted classes. a) 128 HapMap 3 deletion polymorphisms were used to train a RandomForest classifier. The classifier was then tested on HapMap 3 and simulated examples and clearly outperforms on the HapMap 3 examples than the simulated examples. b) 795 simulated deletion polymorphisms were used for training. The classifier was then tested on the two sets, but now outperforms on simulated examples instead of HapMap 3.

<table>
<thead>
<tr>
<th>Tested HapMap 3 set</th>
<th>Trained on HapMap 3 set</th>
<th>hom.</th>
<th>inc.</th>
<th>pA</th>
<th>pB</th>
</tr>
</thead>
<tbody>
<tr>
<td>hom.</td>
<td>17 (1.00)</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td></td>
</tr>
<tr>
<td>inc.</td>
<td>0 (0.00)</td>
<td>48 (1.00)</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td></td>
</tr>
<tr>
<td>pA</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td>26 (1.00)</td>
<td>0 (0.00)</td>
<td></td>
</tr>
<tr>
<td>pB</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td>37 (1.00)</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>17 (0.13)</td>
<td>48 (0.38)</td>
<td>26 (0.20)</td>
<td>37 (0.29)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tested simulation set</th>
<th>hom.</th>
<th>inc.</th>
<th>pA</th>
<th>pB</th>
</tr>
</thead>
<tbody>
<tr>
<td>hom.</td>
<td>81 (0.57)</td>
<td>0 (0.00)</td>
<td>21 (0.15)</td>
<td>40 (0.28)</td>
</tr>
<tr>
<td>inc.</td>
<td>0 (0.00)</td>
<td>190 (0.83)</td>
<td>14 (0.06)</td>
<td>24 (0.11)</td>
</tr>
<tr>
<td>pA</td>
<td>0 (0.00)</td>
<td>1 (0.00)</td>
<td>168 (0.79)</td>
<td>45 (0.21)</td>
</tr>
<tr>
<td>pB</td>
<td>0 (0.00)</td>
<td>1 (0.00)</td>
<td>9 (0.04)</td>
<td>201 (0.95)</td>
</tr>
<tr>
<td>Total</td>
<td>81 (0.10)</td>
<td>192 (0.24)</td>
<td>212 (0.27)</td>
<td>310 (0.39)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tested HapMap 3 set</th>
<th>Trained on simulated chromosomes</th>
<th>hom.</th>
<th>inc.</th>
<th>pA</th>
<th>pB</th>
</tr>
</thead>
<tbody>
<tr>
<td>hom.</td>
<td>8 (0.47)</td>
<td>8 (0.47)</td>
<td>0 (0.00)</td>
<td>1 (0.06)</td>
<td></td>
</tr>
<tr>
<td>inc.</td>
<td>0 (0.00)</td>
<td>48 (1.00)</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td></td>
</tr>
<tr>
<td>pA</td>
<td>0 (0.00)</td>
<td>17 (0.65)</td>
<td>9 (0.35)</td>
<td>0 (0.00)</td>
<td></td>
</tr>
<tr>
<td>pB</td>
<td>0 (0.00)</td>
<td>26 (0.70)</td>
<td>2 (0.05)</td>
<td>9 (0.24)</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>8 (0.06)</td>
<td>99 (0.77)</td>
<td>11 (0.09)</td>
<td>10 (0.08)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tested simulation set</th>
<th>hom.</th>
<th>inc.</th>
<th>pA</th>
<th>pB</th>
</tr>
</thead>
<tbody>
<tr>
<td>hom.</td>
<td>142 (1.00)</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
</tr>
<tr>
<td>inc.</td>
<td>0 (0.00)</td>
<td>228 (1.00)</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
</tr>
<tr>
<td>pA</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td>214 (1.00)</td>
<td>0 (0.00)</td>
</tr>
<tr>
<td>pB</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td>0 (0.00)</td>
<td>211 (1.00)</td>
</tr>
<tr>
<td>Total</td>
<td>142 (0.18)</td>
<td>228 (0.29)</td>
<td>214 (0.27)</td>
<td>211 (0.27)</td>
</tr>
</tbody>
</table>
Table 4.5: SVM had 14 inconsistencies with the HapMap 3 copy number assignments with European trio individuals. A majority of the inconsistencies suppose the deletion called in HapMap 3 has no evidence for deletion occurring in our WGS and HiC read dataset. Indeed, upon manual inspection of the WGS and HiC reads, SVM correctly attributes a strong unlikely deletion score to 8 of the above deletion polymorphisms.

<table>
<thead>
<tr>
<th>ucsc location</th>
<th>Error</th>
<th>Log Prob. Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr14:19269023-19494705</td>
<td>homA-inc</td>
<td>-4.827</td>
</tr>
<tr>
<td>chr6:30569063-30572187</td>
<td>homA-inc</td>
<td>-4.549</td>
</tr>
<tr>
<td>chr7:66258399-66315065</td>
<td>homA-inc</td>
<td>-3.421</td>
</tr>
<tr>
<td>chr13:71374556-71388378</td>
<td>homA-inc</td>
<td>-3.024</td>
</tr>
<tr>
<td>chr13:22529678-22532515</td>
<td>pA-inc</td>
<td>-2.957</td>
</tr>
<tr>
<td>chr3:191219845-191228315</td>
<td>pB-inc</td>
<td>-1.724</td>
</tr>
<tr>
<td>chr5:103881062-103889948</td>
<td>pB-inc</td>
<td>-1.011</td>
</tr>
<tr>
<td>chr15:82330721-82335554</td>
<td>pB-inc</td>
<td>-0.892</td>
</tr>
<tr>
<td>chr4:87194118-87199968</td>
<td>pA-inc</td>
<td>-0.824</td>
</tr>
<tr>
<td>chr5:103881062-103889503</td>
<td>pB-inc</td>
<td>-0.696</td>
</tr>
<tr>
<td>chr2:97506179-97532181</td>
<td>homA-pB</td>
<td>-0.640</td>
</tr>
<tr>
<td>chr14:73067729-73098870</td>
<td>pA-pB</td>
<td>-0.301</td>
</tr>
<tr>
<td>chr5:177159157-177166211</td>
<td>homA-pA</td>
<td>-0.033</td>
</tr>
</tbody>
</table>
4.9.3 Figures

Figure 4.10: Labeling the copy number of a DNA segment in each individual of a trio determines the phase of the deletion in the offspring or homozygosity. Presence or absence of deletion labeling of the mother, father, and offspring cannot determine whether a deletion is homozygous in the offspring. A parent with a deletion present can be either heterozygous or homozygous and may transmit either the deletion or non-deleted allele to the offspring. The left column shows the three patterns of deletion transmittance inferred from copy number. The right column shows the deletion transmittance that can be phased when labeling individuals with deletion presense or absence. HapMap 3 Consortium provided copy numbers for observed deletions. Mill et al. Mills et al. (2011) labeled presence or absence of deletions.
a: WGS 2x b: HiC 2x

**Figure 4.11:** a) Accuracy of simulations with twice the amount of WGS reads. b) Accuracy of simulations with twice the amount of HiC reads.
Figure 4.12: SVM top two scoring inconsistent predictions with HapMap 3 dataset. The WGS and HiC reads strongly support no evidence for a deletion compared to the HapMap 3 homozygous call.

Figure 4.13: Phasing to incorrect error in mills appears in SKA3 gene.
Figure 4.14: Phasing to hom error in real.

Figure 4.15: Difficult to distinguish error.

4.10 Acknowledgements

Chapter 4 is from material submitted for publication, Patel A, Bafna V. *InPhaDel: Integrative whole genome and proximity-ligation sequencing*. The dissertation author was the primary author of this paper, and was responsible for the research.
Chapter 5

Discussion

When I started my thesis in 2009, developing a method to call genome rearrangements from shotgun sequencing experiments was considered passé. Numerous computational tools calling different types of deletions, duplications, and translocations from paired-end read data were already publicly available. Having seen these numerous tools, I assumed the field was saturated and the problems in genome rearrangement research were solved. As this thesis shows, I was mistaken. To summarize my graduate studies, I focused on solving a few of the many open genome rearrangement problems using novel biochemical and sequencing strategies and complementing computational methods.

Firstly, I demonstrated genome rearrangements can be efficiently detected from heterogeneous DNA samples where the mutated DNA represents only a small fraction of the total DNA. I call this assay AmBre (Amplification of Breakpoints) and it could revolutionize cancer monitoring. Genome rearrangements can be used as cancer biomarkers and detected directly from whole blood samples. Clinically, the DNA test would enable determining cancer therapy effectiveness in a matter of weeks rather than months. Also, cancer relapse or recurrence could be captured at
earlier stages of tumor regrowth and better clinical decisions could be made to treat patients earlier. In our publication Patel et al. (2013), I demonstrated AmBre on targeting $CDKN2A$ deletions and $RUNX1-RUNX1T1$ translocations. The targeting of a rearrangement is accomplished by a PCR amplification step to amplify tumor DNA containing the rearrangement breakpoints followed by sequencing of the amplified DNA products to identify the breakpoints. The amplification technique was based on a primer tiling strategy, PAMP (first developed by Liu and Carson (2007)). However, a novel computational algorithm was necessary to design primers to fit the constraints of primer tiling. In addition, to ensure accurate rearrangement calling, we applied long read single molecule sequencing (Pacific Biosciences), which required custom development of another method to detect rearrangements from highly erroneous but long read data. To fully realize AmBre’s clinical utility, the assay needs to be tested on other rearrangement targets besides $CDKN2A$ deletion. In addition, impact of the assay can only be shown by iterative improvements the assay design and testing on clinical samples.

Another area for genome rearrangement research is understanding the biological mechanisms behind rearranging chromosomes. One well established mechanism for cancer genome rearrangement is illegitimate V(D)J recombination, where the machinery responsible for recombining V, D, and J segments in B cells uncontrollably induce rearrangements at alternative V,D,J motifs across the genome. To pin down the mechanism required careful sequence analysis and biochemical studies. In the second chapter of my thesis, I show that sequencing cannot determine whether a rearranged chromosome is formed by progressive rearrangements across numerous cell divisions or a single shattering and repair in a single cell division (chromothripsis). Recent publications have been identifying chromothripsis in rearranged chromosomes using sequencing data, but our analysis reveals the interpretation
could be one or more complex genome rearrangements. The subtle correction is important for the future cancer research. A better definition for complex genome rearrangements will lead to a better understanding of the biological mechanism behind the rearrangement process. Clinically, observing complex genome rearrangement could be used as a prognostic marker. The lasting open questions for the chapter is whether or not “chromothripsis” occurs and if so how can it be detected. Future developments in single cell sequencing and phylogeny analysis of heterogeneous cancer tissues could determine whether the complex genome rearrangements form in a single cell division or across many.

Lastly, I revisit detection of genome rearrangements in the context of a diploid genome. The human genome has 22 pairs of autosomal chromosomes and 2 sex chromosomes, however most genome rearrangement calling tools only predict presence or absence of a rearrangement on 1 of the 23 chromosomes. This assignment is important as recent studies have shown de novo copy number variations are linked increased risk for autism spectrum disorder and schizophrenia. The true genetic cause for disease could be a copy number variation present on a chromosome and rare variant appearing on the same (cis) or homologous (trans) chromosome. To assign rearrangements to chromosomes, I applied techniques developed for the single nucleotide variation (SNV) phasing problem. The SNV phasing problem assigns single nucleotide variations to each homologous chromosome and has been well studied by many computational groups. Recently, whole chromosome phasing was demonstrated by applying novel sequencing strategy based on proximity-ligation Selvaraj et al. (2013). I applied the same concept along with machine learning techniques to demonstrate deletions predicted from whole genome sequencing on a European individual can be accurately phased to each chromosome. The methodology demonstrates a full diploid human assembly
for a single individual could be generated by only shotgun and proximity-ligation sequencing of the individual. To achieve the goal of full diploid assembly, this method needs to be extended to assign duplications and inversions to chromosomes.

Genome rearrangement research is rapidly advancing with upcoming sequencing technologies. While the implementations of computational methods may become obviated in the future, the same computational ideas can still be useful in the analysis of the newer sequencing technologies. As shown in the first chapter, the recently developed Pacific Biosciences single molecule sequencing is capable of highly accurate rearrangement. To facilitate highly accurate calling, we applied concepts from a geometry-based structural variation calling method (GASV Sindi et al. (2009)) to the new form of sequencing data. In conclusion, the methods and findings in this thesis is impactful and can be useful for motivating analysis and interpretations of future sequencing technologies.
Chapter 6

Bibliography


Sridhar Hannenhalli and Pavel Pevzner. Transforming cabbage into turnip (poly-


Yukiko Kitagawa, Kaoru Inoue, Shigeru Sasaki, Yasuhide Hayashi, Yoshinobu Matsuo, Michael R Lieber, Hideaki Mizoguchi, Jun Yokota, and Takashi Kohno.


Jianmin Wang, Charles G Mullighan, John Easton, Stefan Roberts, Sue L Heatley, Jing Ma, Michael C Rusch, Ken Chen, Christopher C Harris, Li Ding, L. Holm-


