ALGORITHMS FOR THE DISCOVERY OF LARGE GENOMIC INVERSIONS USING POOLED CLONE SEQUENCING

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER ENGINEERING

By

Marzieh Eslami Rasekh

July, 2015

Algorithms for the discovery of large genomic inversions using pooled clone sequencing

By Marzieh Eslami Rasekh

July, 2015

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Can Alkan (Advisor)

Assist. Prof. Dr. Ozlen Konu

Assoc. Prof. Dr. Yesim Aydin Son

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

ALGORITHMS FOR THE DISCOVERY OF LARGE GENOMIC INVERSIONS USING POOLED CLONE SEQUENCING

Marzieh Eslami Rasekh
M.S. in Computer Engineering
Advisor: Assist. Prof. Dr. Can Alkan
July, 2015

An inversion is a chromosomal rearrangement in which an internal segment of a chromosome has been broken twice, flipped 180 degrees, and rejoined. Most known examples of large inversions were found indirectly from studies on human disease where inversions have no detectable effect in parents, but increase the risk of a disease-associated rearrangement in the offspring. The development of a map of inversion polymorphisms will provide valuable information regarding their distribution and frequency in the human genome and will help unravel how inversions and the segmental duplications architecture associated with inverted haplotypes contribute to genomic susceptibility to disease rearrangements.

The 1000 Genomes Project spearheaded the development of several methods to identify inversions, however, they are limited to relatively short inversions, and there are currently no available algorithms to discover large inversions using high throughput sequencing technologies (HTS). This is mainly because the breakpoints of such events typically lie within segmental duplications and common repeats, reducing the mappability of short reads.

We propose using pooled clone sequencing (PCS), a method originally developed to improve haplotype phasing, to characterize large genomic inversions. PCS merges the advantages of clone based sequencing approaches with the speed and cost efficiency of HTS technologies. Using this sequencing data, we developed a novel algorithm, *dipSeq* for discovering large inversions (>500 Kbp) following the observation that clones that span the inversion breakpoint will be split into two sections, *split clones*, when mapped to the reference genome.

We evaluate the performance of dipSeq on 3 sets of simulated data, demonstrating its correctness and robustness to structural duplications and other types of structural variations. We further applied dipSeq to the genome of a HapMap individual (NA12878). dipSeq was able to accurately discover all previously known and experimentally validated

large inversions. We also identified a new inversion and confirmed using fluorescent in situ hybridization. Although dipSeq displays a relatively high false positive rate using real data, it performed better with simulated data, suggesting that the performance with the NA12878 genome may be improved with higher depth of coverage.

Keywords: structural variation, pooled clone sequencing, inversion detection.

ÖZET

BÜYÜK İNVERSİYONLARIN TOPLANMIŞ KLON DİZİLEME YÖNTEMİ KULLANILARAK KEŞFİ İÇİN ALGORİTMALAR

Marzieh Eslami Rasekh
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Yard. Doç. Dr. Can Alkan
Temmuz, 2015

Genel olarak kopya sayısı varyasyonu (KSV) ve dengeli yeniden düzenlemeler olarak sınıflandırılabilen çok çeşitli genomik yapısal varyasyon tipleri bulunmaktadır. Her ne kadar literatürde KSV'lerin karakterizasyonu için çok sayıda algoritma varsa da, inversiyon ve translokasyon gibi dengeli yeniden düzenlemelerin keşfi henüz açık bir problemdir. Bunun başlıca sebebi, bu tür varyasyonların kırılma noktalarının parçasal duplikasyonlar ve yaygın tekrarlara denk gelmesi, ve bu durumun kısa okumaların güvenilir şekilde hizalandırılmasını zorlaştırmasıdır. 1000 Genom Projesi inversiyonların bulunması için bazı metotların geliştirilmesine önayak olduysa da, geliştirilen algoritmalar göreceli olarak kısa inversiyonların keşfiyle sınırlıdır, ve büyük inversiyonların yeni nesil dizileme (YND) kullanılarak keşfi için halihazırda bir algoritma bulunmamaktadır. Bu çalışmada, daha önce haplotip haritalama için geliştirilmiş olan bir dizileme metotunu (Kitzman vd., 2011) kullanarak büyük inversiyonların karakterizasyonunu öneriyoruz. Toplanmış klon dizileme adı verilen bu yöntem, klon tabanlı dizilemenin sağladığı avantajları YND teknolojilerinin hız ve masraf etkinliği ile birleştirmektedir. Bu yöntem ile elde edilmiş verileri kullanarak, dipSeq adında, büyük inversiyonları (>500 Kbp) keşfedebilen bir algoritma geliştirdik. dipSeq algoritmasının gücünü önce simüle edilmiş verilerle ispatlayıp, daha sonra da NA12878 kodlu insan DNA'sından elde edilmiş gerçek veriye uyguladık. Bu genomda daha önceden keşfedilmiş ve deneysel olarak ispatlanmış bütün büyük inversiyonları bulabildik. Ayrıca önceden bilinmeyen yeni bir inversiyon polimorfizmini de bulup florasan in situ hibridizasyon yöntemi ile tahminimizi doğruladık.

Anahtar sözcükler: yapısal farklılıklar, toplanmış klon sıralama, inversiyon tespiti.

Acknowledgement

First of all, I would like to express my gratitude to my supervisor, Assist. Prof. Can Alkan, who gave me an opportunity to work in his lab and supported me throughout my masters' studies. I cherish his guidance and patience towards the fulfillment of this thesis.

I acknowledge all the people who collaborated in this work: Assist. Prof. Can Alkan and Assist. Prof. Francesca Antonacci and Prof. Evan E. Eichler for designnig the study, Joyce Tang and Prof. Chris T. Amemiya who built the BAC clones, Prof. Mario Ventura and Assist. Prof. Francesca Antonacci who generated the pooled clone sequencing data, and finally, Giorgia Chiatante and Mattia Miroballo who performed validation experiments. Their efforts are profoundly appreciated.

I thank Assist. Prof. Jacob Kitzman, and Dr. Beth Dumont for data access and their valuable input for the clone reconstruction algorithm, and John Huddleston for computational assistance.

I acknowledge "Marie Curie Career Integration Grant 303772 to Can Alkan" for financial support of this project.

I thank Prof. Claudio Arbib for his valuable discussions on how to formulate the set cover problem with inversions as nodes, Mario Caceres and Sonia Casillas for their help with the InvFEST database, Muhsin Can Orhan for his contributions in the probability calculations for clone reconstruction, Begum Genc and Mecit Sari for their help with the initial formulations of the problem, and Prof. Ugur Dogrusoz for his moral support.

Also, I would like to thank Assist. Prof. Ozlen Konu and Assist. Prof. Oznur Tastan for making my academic life in Turkey more advantageous.

I recall my appreciation to Prof. Naser Nematbakhsh for filling me with optimism in my stressful times.

Finally, I would like to thank my mother and father and especially my sister, Maryam, for her help and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The human genome or DNA consists of about 3×10^9 nucleotides packed into 23 pairs of chromosomes, each containing the coding information necessary for life. From human to human, the genome might slightly differ. Other than base pair mutations called SNPs (single nucleotide polymorphisms), more complex variations might occur e.g. large segments of the genome might be deleted, duplicated, or inverted. Deletions and insertions will cause a loss or a gain and therefore are easier to detect by simply comparing the amount of the genome to the reference. Other types such as *inversions* do not alter the amount of the genome but simply rearrange the order of the genome sequence.

An inversion is a chromosomal rearrangement that occurs when a single chromosome breaks in two locations and rearranges itself such that a segment is reversed and copied back [3]. Inversions usually do not cause any diseases or phenotypical abnormalities in carriers, however, in individuals which are heterozygous for an inversion, there is an increased production of abnormal chromatids which leads to lowered fertility due to production of unbalanced gametes [3] (Figure 2.3).

From a computational perspective, we can rephrase this problem as such: Assume we have a *reference* string of length 3 billion characters and a *donor* string of the same length where some segments of the donor are inverted with respect to the reference. The donor string is hidden from us but we have many short fragments as long as 500 characters from

it. The problem is defined as finding the inverted segments positions.

This problem becomes even more complicated when we know that the strings are composed of 50% repeats and the breakpoints of these inversions are located somewhere on these repeats. With fragments prone to errors and folds smaller than the repeats, how can we detect the inverted segments? Moreover, considering the large size of the inversions (>500 Kbp), many other structural variations such as deletions and duplications might occur close to the breakpoints.

This computational problem has been of interest in modern genomics. The 1000 Genomes Project spearheaded the development of several methods to identify inversions, however, they are limited to relatively short inversions, and there are currently no available algorithms to discover large inversions using high throughput sequencing technologies (HTS). Here we will talk about the motivation and challenges of this work and propose a novel technique to detect large inversions of size >500 Kbp using high throughput sequencing technologies. Further background information and literature study is given in Chapter 2.

1.1 Motivation

Inversions cause normal and disease phenotypic changes and adaptation [4]. Most known examples of large inversions have come indirectly from studies on human disease where inversions have no detectable effect in parents, but increase the risk of a disease-associated rearrangement in the offspring. In the Williams-Beuren syndrome, for example, 25–30% of transmitting parents have a 1.5 Mbp inversion encompassing the commonly deleted region, whereas the same inversion is present in only 6% of the general population [5]. Similarly, a polymorphic inversion has been reported at 15q11-q13 and is seen more frequently in mothers who transmit de novo deletions resulting in the Angelman syndrome [6]. Two more striking examples are found in the Sotos syndrome [7] and the 17q21.31 microdeletion syndrome [8–12]. In each of these disorders, every parent studied to date in which a *de novo* microdeletion arises carries an inversion of the same region. All these inversions

are enriched in segmental duplications at their breakpoints, leading to an increased susceptibility to non-allelic homologous recombination (NAHR) and risk for disease-causing rearrangements to occur in the offspring.

Although there are now many algorithms to discover and genotype structural variation using high throughput sequencing (HTS) data [1, 13], they mainly focus on copy number variants such as duplications and deletions. Balanced rearrangements including inversions are much harder to detect due to the fact that their breakpoints usually lie within complex repeats, reducing mappability. There are very few attempts to characterize inversions and are reliable only for small inversions (∼10–50 Kbp) [14–17], and exhibit high false discovery rates. Only one algorithm, GASVPro [18] is able to detect inversions with a size limit up to 500 Kbp, however its sensitivity and specificity for large inversions are yet untested. Characterization of larger genomic inversions using HTS remains an open problem.

The development of a map of inversion polymorphisms will provide valuable information regarding their distribution and frequency in the human genome and will be important for future studies aimed to unravel how inversions and the segmental duplications architecture associated with inverted haplotypes contribute to genomic susceptibility to disease rearrangements.

1.2 Challenge

Inversions are located in highly repeated regions of the genome reducing the mappability. In large inversions, other structural variations might occur around the breakpoints making the inversion even more complex. In addition, inversions do not alter the amount of the genome and thus, we cannot detect them via read depth signals. In the case of homozygous inversions, where the inversion happens on both strands, de nova assembly cannot help detect the inversion; and in the case of heterozygous inversions, where a normal and an inverted copy of the region, read depth signatures will fail.

The HTS platforms generate data at very high rates with minimal cost. However, since both the HTS reads (100–150 bp for Illumina), and the DNA fragments are very short

(350–500 bp), the mappability of the HTS data is dramatically reduced in repeat-rich regions that harbor most of the inversion breakpoints. On the contrary, the now-largely-abandoned method of clone-by-clone sequencing [19] enables data observation from much larger genomic intervals (40–200 Kbp), but the associated costs are substantially higher.

1.3 Approach

Pooled clone sequencing, a method developed to improve haplotype phasing, aims to combine the advantages of clone-by-clone sequencing, with the cost and time efficiency offered by the HTS platforms [20]. Although pooled clone sequencing was developed to improve haplotype phasing and to characterize large haplotype blocks, we propose a novel algorithm, *dipSeq*, that utilizes pooled clone sequencing to discover large genomic inversions (>500 Kbp).

Our approach to discover large (>500 Kbp) genomic inversion using pooled clone sequencing follows from the observation that, clones (BAC or fosmid) that span the inversion breakpoint will be split into two sections when mapped to the reference genome, also separated by a distance approximately the size of the inversion. We call this sequence signature as *split clones* (Figure 1.1), which is similar to the split read sequence signature used by several SV discovery tools such as DELLY [15] and Pindel [21]. Based on these observations, we developed a novel combinatorial algorithm and statistical heuristics called *dipSeq* (**d**iscover **i**nversions using **p**ooled **Seq**uencing). Briefly, dipSeq searches for both paired read and split clone signatures using the mapping locations of pooled clone sequencing reads, and requires split clones from different pools to cluster at the same putative inversion breakpoints. Ambiguity due to multiple possible pairings of split clones are resolved using an approximation algorithm for the maximal quasi clique problem [22], and paired end read support further assigns confidence score for the predicted inversion calls.

dipSeq proves its potential when tested on simulated data, and it is able to discover previously characterized large inversions (2–5 Mb) in the genome of a human individual (NA12878), using pooled BAC sequence data. dipSeq is *theoretically* compatible with all similarly constructed pooled sequence data, such as the TruSeq Synthetic Long-Reads

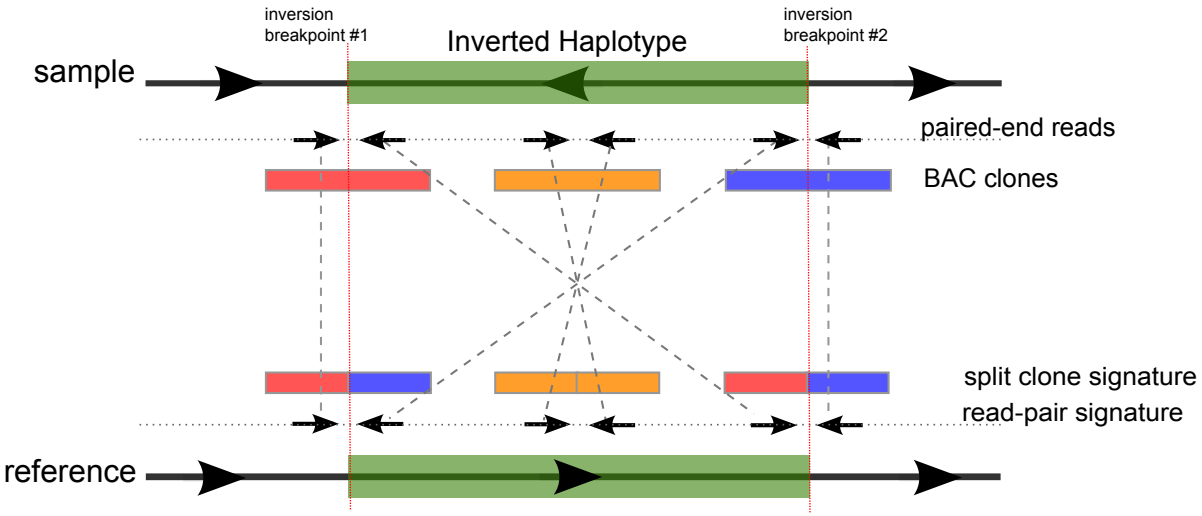


Figure 1.1: Sequence signatures used by the dipSeq algorithm.

(Moleculo) [23], or the Complete Genomics LFR Technology [24], provided that the pooled large DNA fragment sizes follow a Gaussian distribution. However, it should be noted that, large clone size is required to span segmental duplication blocks, and smaller clones such as fosmids may not be sufficient to detect inversions around segmental duplications [20]. Therefore, the theoretical minimum inversion size detectable by dipSeq is limited by clone length, i.e. 150 Kbp when BACs are used.

1.4 Organization of the thesis

This thesis is organized into 6 chapters. In chapter 2, a background on the terminology used in the thesis along with literature study is given. Chapter 3 gives the detailed explanation of our method along with the data preparation and final validation techniques. Testing and simulation results of our work are presented in chapter 4 and chapter 5 focuses on the real data of an individual and novel discoveries. Finally in chapter 6 I will conclude the thesis with a discussion about the advantages and disadvantages of our method along with the future work. More detailed information regarding proofs and parameter adaption of dipSeq is given in Appendix A and B and some statistics of our data is depicted in Appendix D.

Chapter 2

Background information

The whole genetic information encoded as DNA of the human is called the human genome. DNA is a double stranded molecule of nucleic acid sequence packed into 23 chromosome pairs in the cell nucleus. The length of the human genome is more than 3 Gbp, which each bp (base pair) is one of the nucleotides adenine (A), cytosine (C), guanine (G), or thymine (T) and over 50% is repeated. From the computational aspect the genome can be formulated as a long string from the alphabet set of {A, C, G, T}. This long string is stored in 24 chromosomes 1 to 22 and X or Y (only for males). However there are some 'N' characters marking the nucleotides that could not be determined. Not all regions of the DNA are coded and viable. Genes which consist of about 2% of the genome are the main regions to be known to carry the coding information and have functionality. Until now there are an estimated 20,000–25,000 human protein-coding genes [25]. However this number is due to revision and will likely reduce.

2.1 Structural variation

Every human has an unique genome. Including both single nucleotide and structural variations, human genomes are more than 95% similar in between different populations. The

rest is subject to different types of genetic variation [1]. These variants can cause phenotypic differences depending on the regions they affect.

Single Nucleotide Polymorphisms (SNPs) which are mutations of a single nucleotide occur in about 1% of the human population. Each genome is estimated to have over 10 million SNPs. Repetitive SNPs that insert or delete up to 50 base pairs are called InDels.. Microsatelites are repetitions (5-50 times) of few base pairs (2–5 base pairs) .

Recently more studies have been focusing on genomic structural variants, defined as alterations in the DNA that affect >1000 bp that may delete, insert, duplicate, invert, or move genomic sequence [1]. Structural variation (SV) is shown to be common in human genomes [26, 27], which caused increased interest in the characterization of both normal [28–30], and disease-causing large variants [9, 31]. Furthermore, SVs are known to be one of the driving forces of creation of new haplotypes [12], and evolution [32] and thus they can help reveal the evolution path. Different types of structural variations are depicted in Figure 2.1: Imbalanced chromosomal rearrangements or CNVs are gains or losses in the genome. Insertions and duplications cause genomic gain, increasing the amount of the genome while deletions cause losses. On the controversy, balanced SVs such as inversions and translocations do not alter the amount of the genome, which makes the use of read depth signature [13, 33, 34] irrelevant for their detection.

Copy number variations (CNVs) were initially identified using BAC (bacterial artificial chromosome) and oligo array comparative genomic hybridization (CGH) [26, 27, 35, 36], and SNP genotyping arrays [35, 37]. A more detailed map of SV was made possible using fosmid end sequencing [28, 29], however this method was too expensive and time-consuming since it involved creating and plating of fosmid libraries followed with Sanger sequencing. Introduction of HTS data finally made it possible to screen the genomes of many [14, 33, 34, 38] to thousands [30] of individuals.

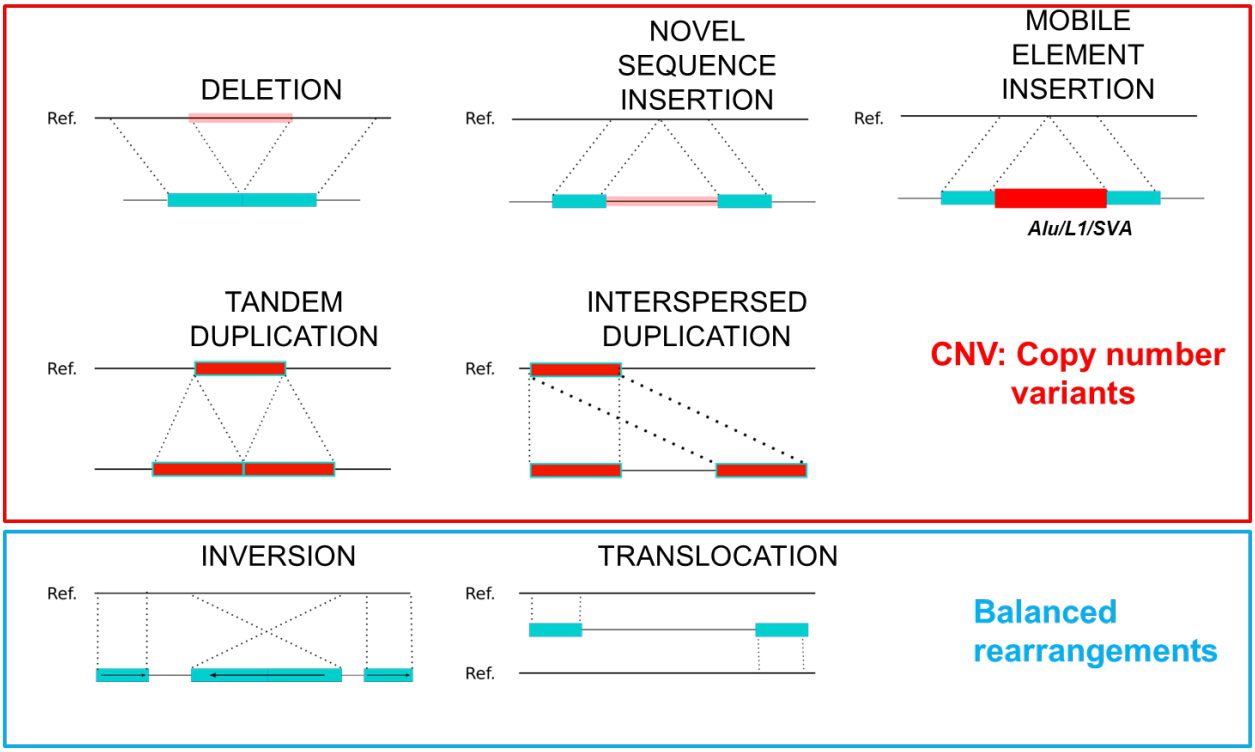


Figure 2.1: Different types of structural variation (adopted from [1])

2.2 Sequencing techniques

In order to understand the genetic divergence of human being we need to read the genome sequence. DNA sequencing is the procedure of finding the letter order of a DNA. Until now the genome has not been fully sequenced and there exists many limitations in the sequencing techniques, namely there is no technique to sequence the genome from start to end and no machine that is errorless. The human reference genome is the most accurate sequencing until now performed on a number of donors. There are different versions of the reference genome. The NCBI36 (hg18) was published on March 2006 followed by the GRCh37 (hg19) edition in Feb 2009 and GRCh38 in December 2013 [39]. In each revision some mistakes due to overlapping repeats were corrected, more individuals were included, and gaps were refined. The hgLiftOver tool from UCSC genome browser [40] can be used to convert different editions to each other. As illustrated in Figure 2.2, there are two major approaches for DNA sequencing:

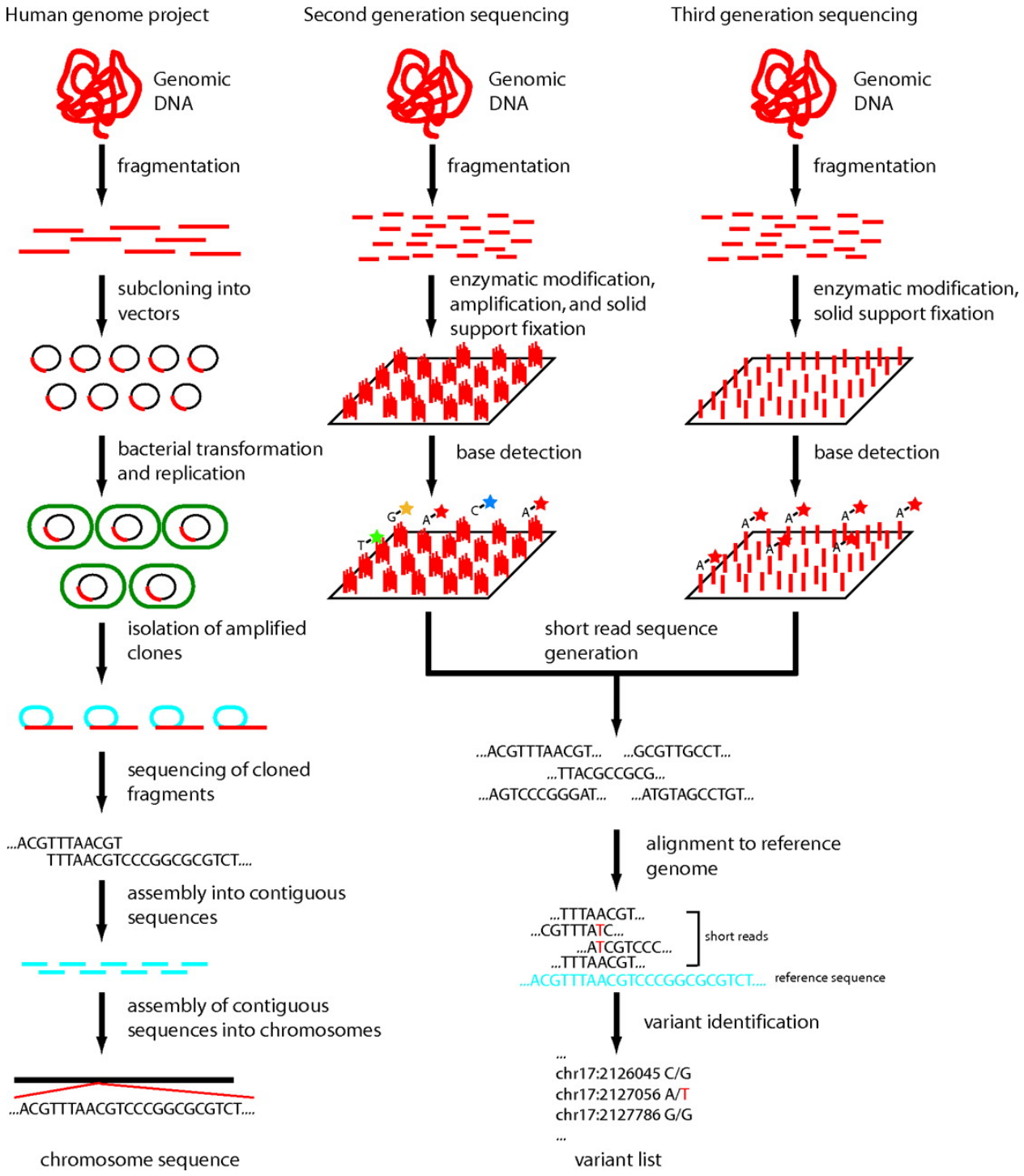


Figure 2.2: Different techniques used for DNA sequencing (adopted from [2])

2.2.1 First generation sequencing

Gilbert and Sanger were the first to introduce two similar methods to sequence the human genome. In the Sanger method, which became more common, DNA is sheared into large clones. Each clone is labeled and replicated separately then sheared into small fragments which are later sequenced. The small sequences are assembled in a hierarchical approach to construct contiguous larger sequences, and then reordered to infer the original clones, and given the clone orders, finally the chromosomes can be built. This method was used to sequence the human reference genome due to its high accuracy and very low error rate despite its extreme cost and time requirements. The Sanger method was later improved by fixing the DNA molecules into a matrix called clusters, automatizing the sequencing into a machine and making it much faster.

2.2.2 Next generation Sequencing

In the second or next generation sequencing (NGS), all DNA is sheared into random small fragments and the fragments are read from one or both end to produce short reads making high throughput sequencing (HTS) possible. These reads are usually <1000 bp long. The assembly and reconstruction of the genome requires more computational effort due to high error rates and repeats which are mostly larger than the short reads. However, because this technique is much cheaper, more DNA can be sequenced proving higher coverage and read depth. In theory NGS is capable of reconstructing the whole genome and detecting any SVs, however until now, this ambition remains far from fulfillment.

2.3 Inversions

An inversion is a chromosomal rearrangement in which an internal segment of a chromosome has been broken twice, flipped 180 degrees, and rejoined [3]. They are mostly viable and will cause a disease only if the breakpoints are located on genes, and otherwise simply

change the rearrangement of genes. Inversions can be heterozygous or homozygous. In heterozygous inversions, due to the loop produced (Figure 2.3), crossovers will lead to lethal product and so, the inversion region will have a lower recombinant frequency forcing the recombinant factor of the genes inside the inversion loop to be zero [3].

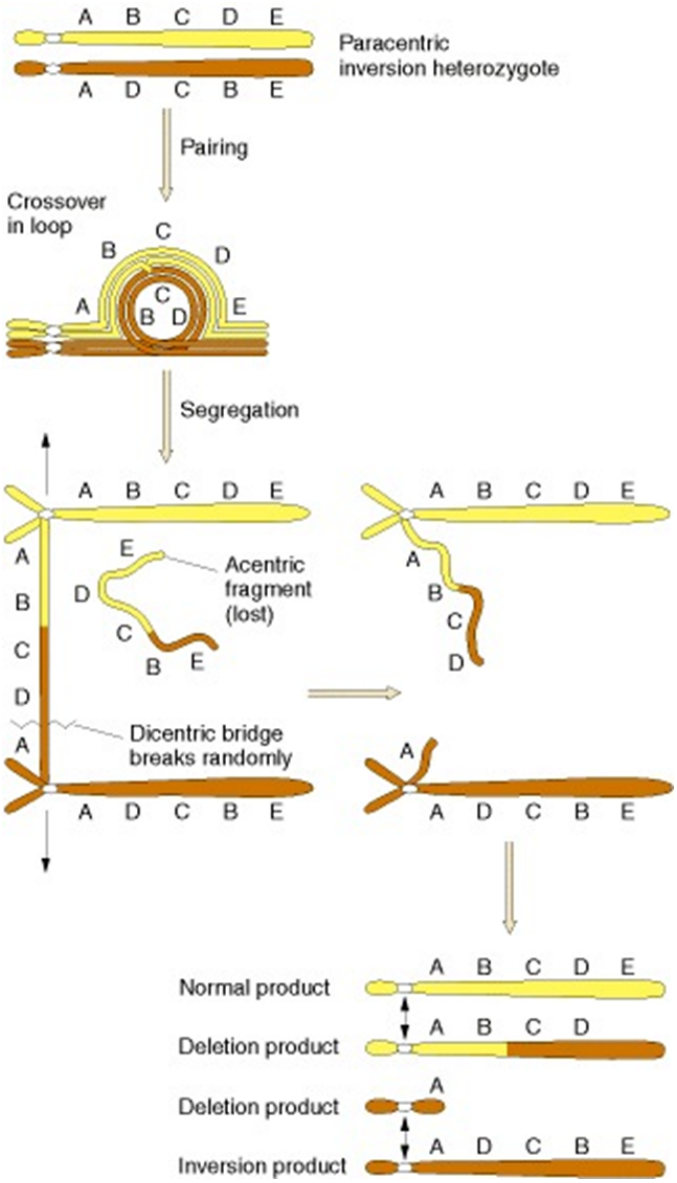


Figure 2.3: Meiotic products resulting from a single crossover within an inversion loop (adopted from [3])

2.3.1 InvFEST

InvFEST is an open source database available online that stores all predicted and validated human polymorphic inversions in the literature [41]. The dataset provides inversion breakpoint coordinates of different healthy individuals based on the March 2006 human reference sequence (NCBI Build 36.1, hg18). This web service, the only comprehensive resource for inversions, is provided with a strong search engine to search and query data. Also the database can be downloaded and accessed offline for more user specific queries. InvFEST includes studies performed up to 2013 and is now incorporating some of the studies performed in 2014 and 2015. As it can be observed in Table 2.1, only 2.63% of the inversions reported in the literature are larger than 500 Kbp.

Table 2.1: Inversion statistics in InvFEST

inversion status	total/hg18	breaks genes	size <500 Kbp	size >500 Kbp	breaks gene and size>500 Kbp
predicted	532	48	517	5	7
unreliable prediction	424	69	416	15	2
validated	86	7	80	8	1
FALSE	50	–	51	0	–
total count	1092	124	1064	28	10
percentage	100.00%	11.36%	97.44%	2.63%	35.71%

Predicted means the inversion has been predicted by at least one study. Validated means they inversion was validated and confirmed in at least one study and one individual. FALSE means the inversion was not validated on any individual.
However, these numbers are not reliable since InvFEST is redundant. In general only 16 validated inversions of size >500 Kbp are reported in InvFEST.

2.4 Detecting inversions using HTS data

Many tools have been implemented in the literature to detect inversions which all use 5 basic approaches to detect inversion signatures using HTS data.

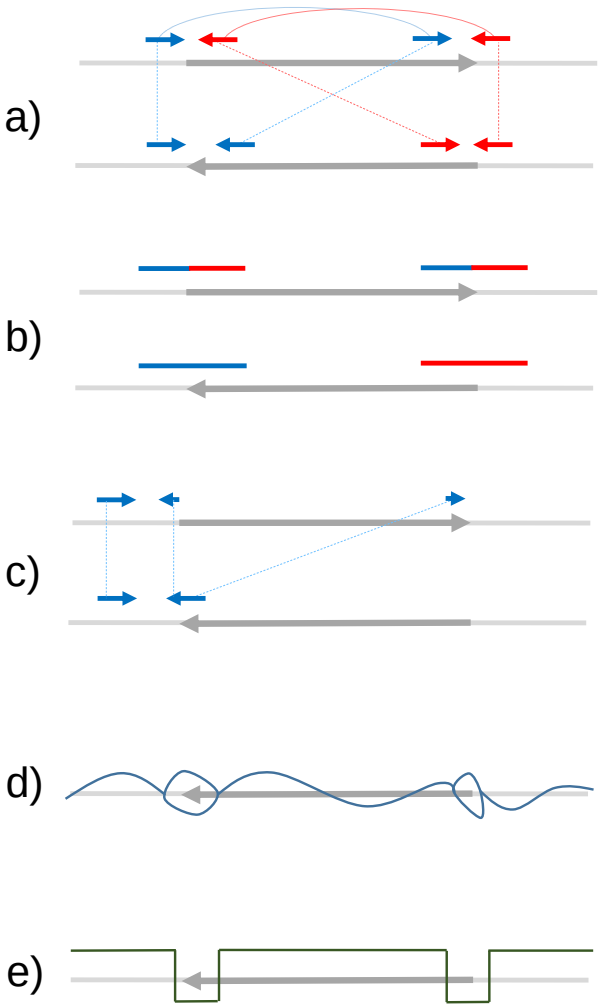


Figure 2.4: Sequencing signatures used to detect inversions

a) Paired read signature: paired reads located near the breakpoints will map on the same strand in a distance same as the inversion. b) Split read signature: the split read will map in the distance of the inversion. c) Soft clipping: if a paired end is located exactly on the breakpoint, the CIGAR value in the BAM file will indicate that the read was mapped up to which base pairs and the rest was unmapped. d) de Bruijn graphs: in de nova assembly, by prefix-postfix matching, a bubble will be observed in the graph. e) Read depth: looking at concordant reads, on the breakpoints less reads will map making a fall in the read depth.

Table 2.2: Available tools to detect inversions using HTS data

Tool	**Data required**	**Size range**	**Signal**	**Year**
AGE [42]	split reads	given breakpoints	*	2011
BreakDancer [43]	paired end reads	>10 Kbp	1	2009
BreaKmer [44]	paired end reads	genotyping	4	2014
ClipCrop [45]	paired end reads	<5 Kbp	3	2011
Cortex [46]	paired end reads	<10 Kbp	4	2012
CREST [47]	paired end reads	(×)	4	2011
Pindel [21]	paired end reads	given breakpoints	4	2009
GASVPro [18]	paired end reads	<500 Kbp	1, 5	2012
Gustaf [48]	single end reads or paired end reads	<5 Kbp	2,3	2014
inGAP-sv [49]	paired end reads	<1 Kbp	1, 5	2011
INVY [15]	paired end reads and split reads	<5 Kbp	1, 2	2012
LUMPY [16]	paired end reads or/and split reads	<10 Kbp	1, 2, 3, 5	2012
Meerkat [50]	paired end reads	<10 Kbp	1, 3	2013
MetaSV [51]	paired end reads	(×)	1, 2, 3, 5	2015
PEMer [52]	paired end reads	<10 Kbp	1	2009
PRISM [53]	paired end reads	(×)	1	2012
SHEAR [54]	paired end reads	<30 Kbp	4	2014
SOAPsv [55]	paired end reads	<50 Kbp	4	2011
SoftSearch [56]	paired end reads	<50 Kbp	1, 2	2013
SVDetect [57]	paired end or mate pair	(×)	1	2010
SVMiner [58]	paired end reads	<100 Kbp	1	2012
TakeABreak [59]	paired end reads	<2 Kbp	4ˆ	2014
TIGRA [60]	paired end reads	given breakpoints	1,2, 4	2015
VariationHunter [61]	paired end reads	10 Mbp	1	2010

* Performs fine aligning to find the exact position of the breakpoint.
(×) Not tested or mentioned.

1. **Paired read signature**: The most common method to discover inversions is to analyze the read pair signature [1, 13], where the mapping strand of the read pairs spanning the inversion breakpoints will be different from what is expected (Figure 2.4a). For example, the Illumina platform generates read pairs from opposing strands, however, if the DNA fragment spans an inversion breakpoint, they will both be mapped to the same strand. They will also be separated from each other by a distance approximately same with the inversion size. When the inversion is large, the *real* mapping distance between pairs also increases, therefore increasing the chance of incorrect mapping due to the common repeats and segmental duplications near the breakpoints.

2. **Split read signature**: If a split read spans an inversion breakpoint, the two splits will map in a distance larger than expected [1] (Figure 2.4b). Split read signatures can be useful for small insertions and deletions, but in the case of large inversions due to the segmental duplications at the breakpoints, the splits will not precisely map to the inversion breakpoint. Also split reads mapping techniques have a limited search space and will try to map only few standard deviation away. However, once we know the approximate breakpoints, this method is useful to refine the breakpoints found by paired reads.

3. **Soft clipping**: Another similar signature to split reads is soft clipping. In this case looking at the CIGAR value of the BAM file containing the paired end reads mapping on the same strand (the first signature), the clip point is extracted and used as the predicted breakpoint (Figure 2.4c). Note that in such a case, reads with low map quality should be included which makes the signature sensitive to noise and SNPs; thus, this method cannot be used alone without further improvement.

4. **de Bruijn graphs**: Another approach suggested in the literature is *de novo* assembly and use of de Bruijn graphs. Each SV type will produce a unique bubble signature, and inversions make a forked loop (Figure 2.4d). This method can be useful in the case of simple and small genomes. As the genome gets larger and more complicated, such as the human genome, more computational power and memory is required. However in the case of genomes that the reference is not available or poorly

assembled, this signature can be useful [59]. Also, in the case of genotyping inversions (i.e. if the approximate breakpoints are given), de nova assembly can be applied to refine the breakpoints.

5. **Read depth signature**: Although more commonly implied to CNVs, in few tools read depth signals from concordant reads have been used to detect inversions. Given the concordantly mapping reads, at the breakpoints of an inversion the read depth will decrease relatively due to unmapped reads (Figure 2.4e). This signature is very noisy and cannot be directly used to detect inversions. Especially in large inversions, deletions may happen inside the breakpoints misleading the algorithm to detect it as a breakpoint.

In practice, due to mapping errors and complexity of inversion regions, no one approach can precisely define an inversion. Most tools incorporate further techniques to discard false calls from the true ones. Others use multiple approaches to find reliable inversion calls. A list of available computational tools that can detect inversions are listed in Table 2.2. The inversion size given in the table indicates the largest inversion size that has been tested or claimed by the authors. As it can be observed most tools fail to find large inversions. Also most tools have high false positive rates.

GASVPro [18] is the only tool able to detect inversions with a size limit up to 500 Kbp, however its sensitivity and specificity for large inversions are yet untested. In their algorithms read depth signature from concordantly mapped reads supported by paired read signatures are extracted and "utilizes a Markov Chain Monte Carlo procedure to sample over the space of possible alignments". Most recently, LUMPY [16] was developed, which integrates multiple sequence signatures, including read alignments, and prior knowledge into a probabilistic framework and has been tested on inversions up to 10 Kbp. BreakDancer [43] and VariationHunter [61] can potentially find large inversions but they have not been tested on large inversions so far. BreakDancer extracts regions encompassing paired read signatures statistically more than the average and uses a consensus to assign the type of the SV and calculated a confidence score. As mentioned by authors, out of the 4 inversions they simulated with size <8 Kbp, only 3 were called. VariationHunter used paired read signatures, and aims to cluster all the signaling paired ends by solving the maximum set cover problem. SVDetect claims to find inversions of arbitrary size but

has not performed any simulation tests and did not call any inversions on the NA12878 individual. CREST [47] could identify one inversion >100 Kbp. INVY from the DELLY package [15] uses uniquely mapped pair ends to find paired read inversion signatures, clusters them, then tries to refine the breakpoints using split read signatures in the same region. AGE [42], Pindel [21], and TIGRA use *de novo* assembly to refine given ambiguous breakpoints using poorly mapped reads (low quality score) or orphan pairs (one pair not mapping). MetaSV and SHEAR combine calls from several other standalone tools and make a consensus. All the aforementioned tools have limitations on the inversion size and although their underlying techniques and mostly similar, there is little overlap between different tools as each is optimized for specific data and purposes [1].

2.4.1 Validation and genotyping

Once an inversion has been predicted, different methods can be applied to validate the genotype in the laboratory. Hybridization-based microarrays picture copy number gains and losses of the donor genome in compare to the reference and therefore are not useful to genotype inversions. For validating inversions, visualization at the single-molecule level should be used such as fluorescent in situ hybridization (FISH), fiber-FISH and spectral karyotyping which were previously used to identify large multi-chromosomal duplications [62]. Although limited due to their low throughput and low resolution, these methods can be applied to large structural differences (∼500 Kbp to 5 Mbp). Metaphase fluorescent in situ hybridization (FISH) can validate inversions larger than 2 Mbp using two probes located inside of the inversion and looking at their relative position. Similarly, interphase triple-color FISH can validate inversions smaller than 2 Mbp and larger than 500 Kbp using two probes inside and one outside the inversion.

Chapter 3

Methodology

In summary, dipSeq discovers inversion polymorphisms in a high-throughput approach by taking advantage of a recently developed method which enables experimental haplotyping of whole genomes [20]. This sequencing method is briefly described in Section 3.1. dipSeq is applied to the provided sequencing data in a multistep fashion. The details of dipSeq is explained in Section 3.3. Further discussion on parameter tuning, compatibility, and restrictions of dipSeq is presented in Chapter 6.

3.1 Pooled clones sequencing data

dipSeq uses pooled sequencing data. This data consists of a number of clones from the genome with average length and a reasonable standard deviation into a number of pools. Each pool is then separately sequenced using any HTS sequencing technique to produce the fastq files required as input for dipSeq. The advantage of this data is that is benefits from the advantages of clone-based sequencing while is cost is relatively low due to the HTS approach. We are interested in the fact that when clones are randomly divided into several pools, the probability of having overlapping clones in each pool will be relatively low. Thus, given that the clones come from a Gaussian distribution with a given lower cutoff, using a simple sliding window approach and extending reads we can reconstruct the

clones of each pool using the HTS data after mapping the reads. The overall procedure is depicted in Figure 3.1.

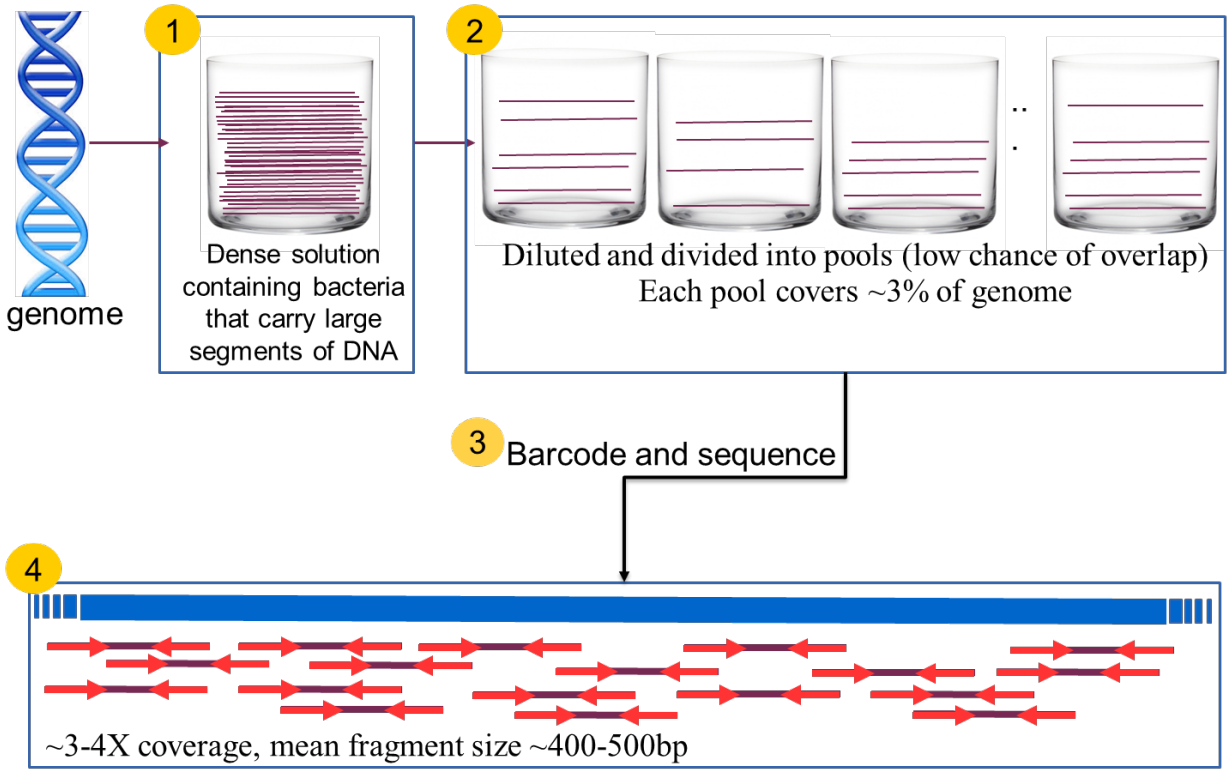


Figure 3.1: Pooled clone sequencing.

3.2 Read mapping

We first map the paired-end reads generated for each pool separately to the human reference genome assembly. Our dipSeq algorithm does not depend on any specific aligner, but in this study we used both BWA [63], and mrFAST [64]. We then separate the read pairs that map in the same orientation (i.e. paired read signature for inversions using Illumina), and those that map concordantly within 4 standard deviations of the average fragment span size into separate files to facilitate easier clone reconstruction and read pair support calculation described in the following sections.

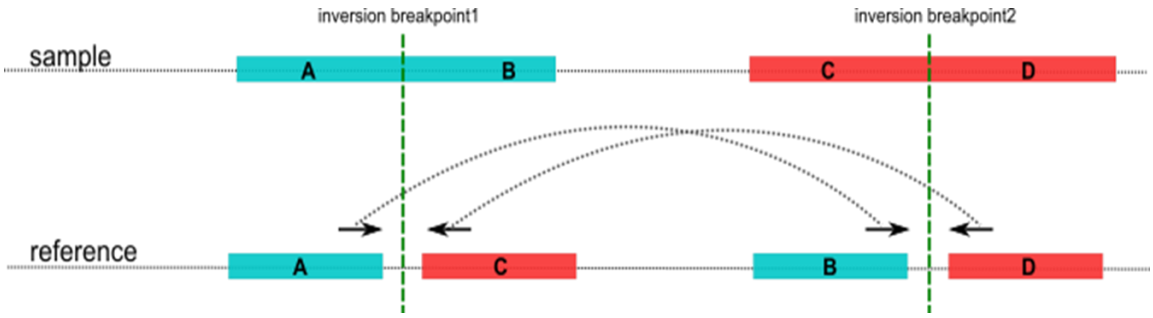


Figure 3.2: Sequencing signatures used by dipSeq to detect large inversions.

Clones spanning the breakpoints will break into two split clones in the distance of the inversion (split clone signature) and the paired end reads from fragments spanning the breakpoints will map to the same strand in the distance of the inversion (paired end signature).

3.3 dipSeq algorithm

dipSeq is based on the basic idea that if a clone spans the inversion breakpoints, when reconstructing it from the mapped reads, we will observe two broken clones called split clones as illustrated in Figure 3.2. Using this signature, along with the paired read signature we can detect the inversions.

dipSeq takes a number of mapped paired end reads in the format of BAM files as input. The parameters to set are the minimum and maximum inversion size. The rest of the parameters are calculated from the data (Section 3.5). The algorithm proceeds by extracting the information it needs for the future steps from the BAM files. Using concordant reads initial clones are reconstructed (Section 3.3.1) and used to detect paired split clones (Section 3.3.2). Potential inversions are made by clustering two compatible paired split clones (Section 3.3.3). The inversions are further clustered to refine the breakpoints using a quasi-clique algorithm (Section 3.3.4). The final inversion clusters along with support information is given as output in a tab separated file (tsv). The overall algorithm of dipSeq is illustrated in Figure 3.3 and explained step by step in the following sections.

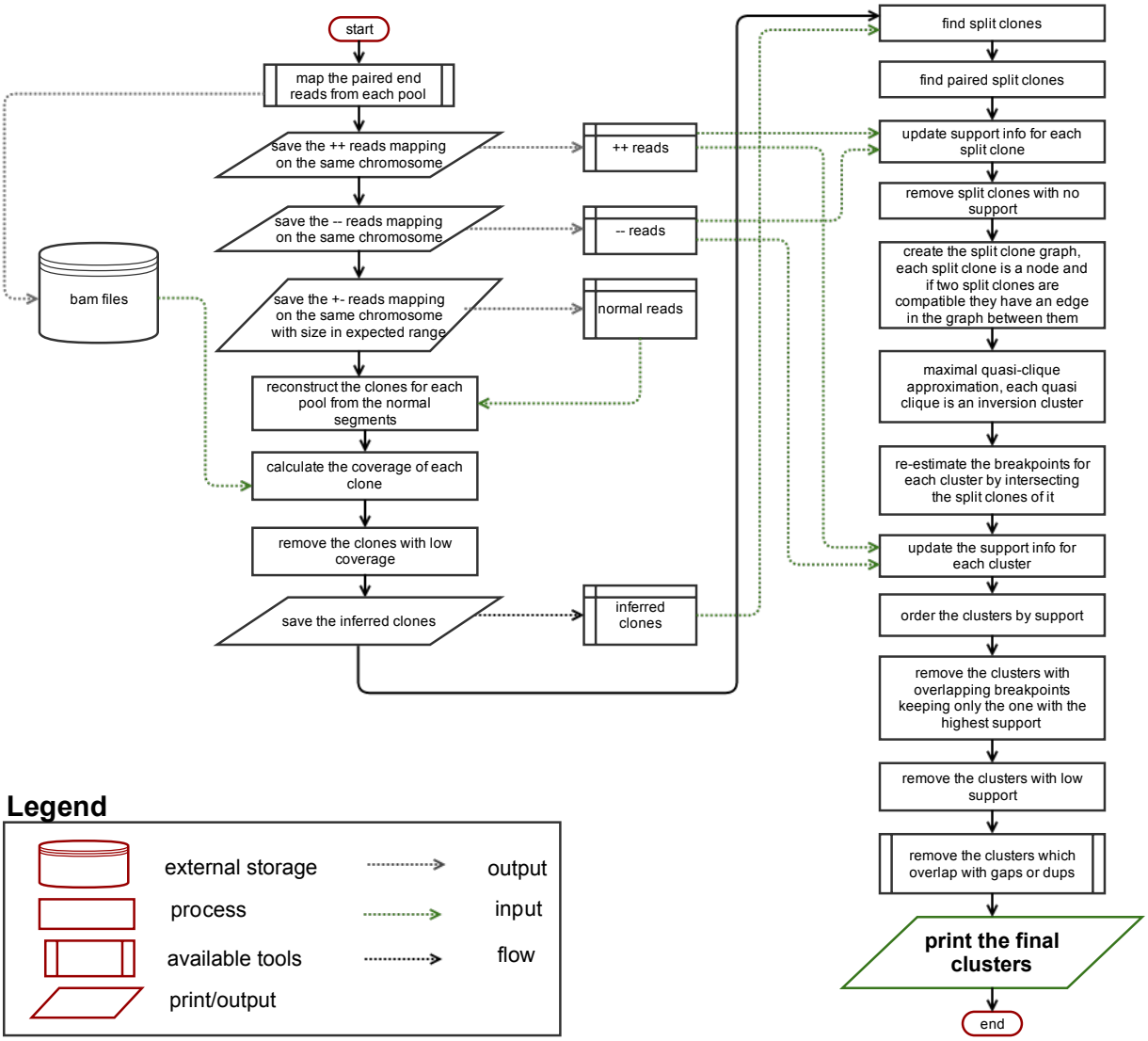


Figure 3.3: Overview of the dipSeq algorithm.

3.3.1 Reconstructing clones

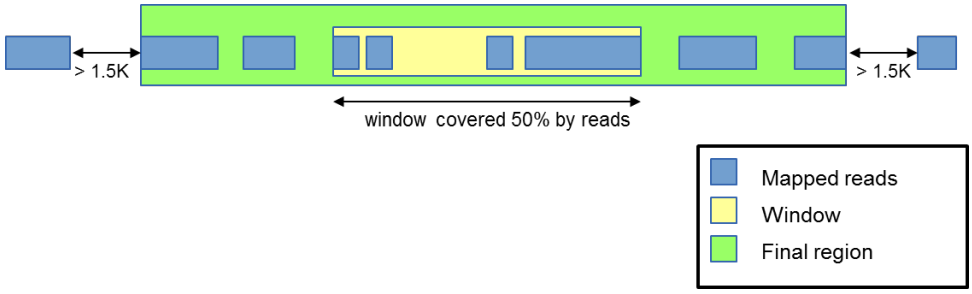


Figure 3.4: Reconstructing the clone using only the concordant reads.

dipSeq uses only the concordantly mapped read pairs to infer the locations of clones. However, due to the low depth and breadth of coverage, it is not always possible to observe a continuous mapping of read pairs that collectively span genomic intervals within expected size of BAC clones. To overcome this issue, we apply several heuristics to identify clone locations. As illustrated in Figure 3.4, scanning from the beginning to end of each chromosome's reads, we first identify windows that are covered by at least 50%. We use such regions as seed windows and then extend these windows using any read pairs that map to its flanking regions with a distance of at most one average fragment size (calculated from the data). Although the parameters we used here may seem arbitrary, in fact they were obtained by applying an optimization grid on simulated BAC data given in Appendix B. This algorithm runs in $O(n \log n)$ time for sorting the reads, and amortized run time of $O(n)$ for reconstructing the clones, where n is the number of paired end reads.

3.3.2 Paired split clones

In the next step we search for paired split clones in each pool. This is done by searching for split clones (clones that are smaller than the average size) that if paired the summation of their lengths will be within an expected size range of $\mu_{\text{clone}} \pm 3\sigma_{\text{clone}}$ where μ_{clone} is the mean clone size (i.e. $\sim$150 Kbp for BACs) and σ_{clone} is the standard deviation. We also require the distance between the split clones to be within the inversion size limits we are trying to discover. Therefore, two regions r_k and r_l are predicted to be a paired split clone,

denoted as $\text{PSC}_{\text{r}_\text{k},\text{r}_\text{l}}$ if:

$$\mu_{\text{clone}} - 3\sigma_{\text{clone}} \leq |\text{r}_\text{k}| + |\text{r}_\text{l}| \leq \mu_{\text{clone}} + 3\sigma_{\text{clone}}$$

$$\text{min\_inv\_size} \leq |\text{r}_\text{l}.\text{start} - \text{r}_\text{k}.\text{end}| \leq \text{max\_inv\_size}$$

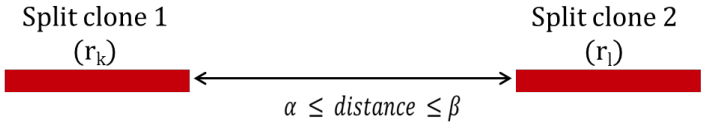


Figure 3.5: Paired split clone.
α and β are the minimum and maximum inversion size

This procedure is depicted in Figure 3.5. Theoretically dipSeq can detect any inversion larger than one clone size.

Assuming the inferred clone locations are sorted by mapping locations, our algorithm can detect split clones in $O(n)$ amortized run time, where n is the number of inferred clones. However, the constant coefficient increases rapidly with the increase of average read coverage.

3.3.3 Inversion Clusters

Next we try to combine two compatible paired split clones to detect a potential inversion. Note that the paired split clones should come from different pools and should be compatible (i.e. same breakpoint locations and inversion size). We denote such compatible pairs as an inversion cluster.

The conditions of combining two paired split clones is illustrated in Figure 3.6. Due to both mapping errors and biases caused by our sliding window approach, we permit a gap or overlap between the paired split clones (Figure 3.6). We expect the inversion breakpoints to lie between these gaps. Two paired split clones $\text{PSC}_{\text{r}_\text{k},\text{r}_\text{l}}$ and $\text{PSC}_{\text{r}_{\text{k}'},\text{r}_{\text{l}'}}$ are *compatible* to be in the same paired split clone (PSC) set, assuming $\text{r}_\text{k}/\text{r}_{\text{k}'}$ are located upstream of $\text{r}_\text{l}/\text{r}_{\text{l}'}$, if:

$$\text{max\_overlap} < \text{r}_{k'}.\text{start} - \text{r}_{k}.\text{end} < \text{max\_gap}$$

$$\text{max\_overlap} < \text{r}_{l'}.\text{start} - \text{r}_{l}.\text{end} < \text{max\_gap}$$

Here we set the max_gap $= -1 \times$ max_overlap $= \mu_{\text{clone}}$. Adding more split clones to the same cluster will narrow down the gap size in breakpoint estimate.

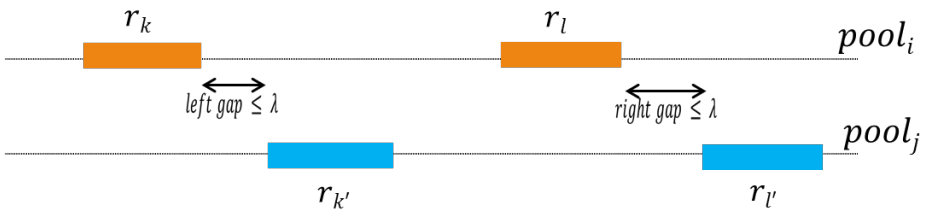


Figure 3.6: An inversion cluster.

However, not all of the inversion clusters we identify signal a real inversion event. In an ideal case where there are no mapping errors, other forms of structural variation, or areas with low mappability may cause paired split clones signatures which might be mistakenly included in an inversion cluster. To ensure only true inversions are detected, we also require read pair support for inversions [1, 13], and we discard any inversion cluster that are not supported by read pairs. This step of the algorithm runs in $O(m + n)$, where m is the number of read pairs with inversion signature and n is the number of split clones given than $m \gg n$.

3.3.4 The inversion graph

Each inversion cluster gives a hint about the existence of an inverted haplotype. However, a paired split clone may have multiple potential "mate"s with similar properties, and therefore be present in multiple inversion clusters. Also some inversion clusters might be supported by paired read signatures coming from sequencing noise or mapping errors. To both resolve ambiguities from multiple possible paired split clone combinations, and unambiguously identify inversions, we construct an undirected graph, where each inversion cluster is a node, and an edge between two nodes indicates that share predicted breakpoints.

We initially formulated this problem as a set cover problem or the equivalent maximum clique problem similar to VariationHunter [61], however, we observed in both simulation and real data sets that due to segmental duplications and deletions around the breakpoints, set cover approximation selected only one of the inversion breakpoints correctly. We therefore formulate the problem as finding maximal quasi-cliques in the inversion graph. This formulation allows existence of incomplete cliques, and tolerates some split clones to be included in a true cluster, and as a result, increases flexibility and avoids getting stuck in a local optimum.

We construct a graph $G = (V, E)$ as follows. Each node in the graph denotes an inversion cluster, as explained above, and each node will therefore represent a pair of regions. We put an edge between two nodes if the two representative inversions agree with breakpoint locations through simple intersection (they are compatible with each other). Formally,

$$V = \{v_i \ : \ v_i \text{ denotes an inversion cluster}\}$$

$$E = \{(v_m, v_n) \ : \ \text{breakpoints}(v_m) \cap \text{breakpoints}(v_n)\}$$

To find an approximate solution for the maximal quasi-clique problem, we use an approximation algorithm previously suggested by Brunato et al [22]. By definition given parameters λ and γ where $0 \leq \lambda \leq \gamma \leq 1$, the subgraph induced by the node set $V' \subseteq V$ is a (λ,γ)-quasi-clique if and only if:

$$\begin{cases} (A) & \forall v \in V' : deg_{V'}(v) \geq \lambda(|V'| - 1) \\ (B) & |E'| \geq \gamma.\binom{|V'|}{2} \end{cases}$$

where $E' = E \cap (V' \times V')$. This means each node $v \in V'$ is connected to at least $\lambda.|V'|$ other nodes and the ratio of edges present in the (λ,γ)-quasi-clique to a complete clique of the same size is γ.

The approximation algorithm starts by sorting all the nodes according to their degrees and takes the node with the largest degree as the initial quasi-clique node set V'. In each run, one node is removed and another is added ensuring that conditions (A) and (B) are not violated. The steps of each iteration are described below.

1. **Adding a node :** First we should make a set of critical nodes defined as the set of nodes in V' that if a new node which is not connected to them is added to V', they will no longer satisfy condition (A). Formally said:

   $Crit(V') := \{v \in V' : deg_{V'}(v) < \lceil \lambda.|V'| \rceil\}$

   Then we make a set of nodes eligible to add to V' as:

   $Add(V') := \{v \in V \backslash V' : deg_{V'}(v) \geq \max\{\lceil \lambda.|V'| \rceil, d_{V'}\} \wedge \{v\} \times Crit(V') \subseteq E\}$

   where $d_{V'}$ is the global density constraint for adding a new node making sure condition (B) is not violated defined as $d_{V'} := \lceil \gamma.\binom{|V'|+1}{2} \rceil - |E'|$.

2. **Removing a node :** A node that is connected to all nodes in the RCrit set is eligible for removal from V' because by losing one edge they would no longer satisfy condition (A).

   $RCrit(V') := \{v \in V' : deg_{V'}(v) - 1 < \lceil \lambda.(|V'| - 2) \rceil\}$

   The set of edges that can be removed to improve the quasi-clique without violating condition (B) is defined as:

   $Rem(V') := \{v \in V' : (\{v\} \times RCrit(V')) \cap E = \emptyset \wedge deg_{V'}(v) \leq e_{V'}\}$

   where $e_{V'}$ is the global density contraint for removing a node without violating condition (B) and is defined as

   $e_{V'} := |E'| - \lceil \gamma.\binom{|V'|-1}{2} \rceil$.

3. **Plateau moves :** A plateau move is a node removal followed by a node addition without violating conditions (A) and (B).

   $PAdd(V') := \{v \in V \backslash V' : deg_{V'}(v) \geq \lambda.(|V'| - 1)\}$

   Note that $Add_{V'} \subseteq PAdd_{V'}$ and once $w \in PAdd_{V'}$ is chosen, $V' \cup w$ might violate the conditions (A) and (B). Thus we define a set of plateau critical nodes that would violate the condition if removed:

   $PCrit(V', w) := \{v \in V' \cup w : deg_{V'}(v) - 1\}$

   When removing a node we must make sure it is not connected to a plateau critical node which results is losing too many edges from $V' \cup \{w\}$. Maximum number of edges we can afford to lose is:

   $r_{V',w} := |E'| + deg_{V'}(w) - \gamma.\binom{|V'|}{2}$

Thus the set of plateau removable nodes would be:

$$PRem(V', w) := \{v \in V' : deg_{V' \cup \{w\}}(v) \leq r_{V',w} \wedge (\{v\} \times PCrit(V', w)) \cap E = \emptyset\}$$

In each iteration one node is removed followed by an addition (first node in list since they are sorted by degree, $O(1)$). When removing and adding nodes, in order to not get stuck in a repetition, a new variable is introduced called *tabu* to ensure that nodes are not added and removed more than *tabu* times. If no more nodes can be added or removed, the algorithm terminates returning V' as the maximal quasi-clique. Proof of the algorithm is given in [22].

We set the *tabu*, γ, and λ parameters to $log(|V|)$ rounds, 50%, and 60%, respectively. The values for these parameters were obtained by a grid optimization on experimental graphs depicting worst case scenarios (see Section 3.5). The graph was implemented using the *Set* object of Java.

dipSeq runs the maximal-quasi-clique algorithm over and over, until no further cliques can be found. Every time a quasi-clique is found, we remove the paired split clones inside the nodes, resulting in the removal of any inversion that was based on those paired split clones. We do not remove the paired read signature at this point due to later breakpoint refinement. In this step we can see the power of maximal quasi-clique in compare to the maximal set cover (equivalent to maximal clique) formulation. In the former we have more freedom to find larger cliques which are missing some edges. But the later would return smaller and complete clique which are usually due to repeats near the breakpoints. Using the maximal set cover approximation we could find only one previously known inversion on the real data but the maximal quasi-clique implementation returned all. More discussion on the differences is given in Appendix A.2.

The complexity of this algorithm is not provided by the authors. However it is simple to see, each iteration takes $O(|V|)$ and maximum tabu$^2O(|V|)$ iterations are required. Since inversions are ordered according to position, an extra amortized cost of $O(|V|)$ is required to remove the nodes and a maximum of $O(|V|)$ quasi-cliques might exist. Thus the overall complexity is $O(n^3)$.

After finding the quasi-cliques we refine the breakpoints with basic intersection. Due to

the overlaps we allowed, not all inversions in a quasi-clique will intersect with each other. We form another graph and look at the intersection of as many inversions possible. Meaning we exclude the breakpoint that agrees with less nodes.

Next, the paired read support for the breakpoints of the final quasi-cliques within the distance of one fragment size is recalculated using the discordant read pairs. The reason we allow for some distance is that reads on the exact breakpoint would not map correctly. We report the final clusters after removing those that intersect with known duplications and assembly gaps (>40%).

3.4 Output format

The final breakpoints are output in a tab separated file (tsv) given the following fields:

1. chromosome
2. left breakpoint start position
3. left breakpoint end position
4. right breakpoint start position
5. right breakpoint end position
6. sum of the paired read support of the inversion clusters (+/+)
7. sum of the paired read support of the inversion clusters (–/–)
8. number of paired split clones supporting the breakpoints
9. number of the paired read support of the refined breakpoints (+/+)
10. number of the paired read support of the refined breakpoints (–/–)

Also dipseq gives all the paired split clones supporting the breakpoints in separate files.

3.5 Parameters

User specific parameters of dipSeq are the input bam files as input, and the minimum and maximum inversion size. The optional parameters are: file fixing (we used this for filtering DIVET files provided by the mrFAST aligner), and chromosome (to specify a special chromosome to run). Other parameters are calculated from the data. The complete list of parameters used by dipSeq is given in Table 3.1.

Table 3.1: dipSeq parameters.

Paired-end read information		
Parameter	**Explanation**	**Value**
READ_LENGTH	Length of each read	*from data*
FRAG_MAX	Maximum fragment size from the paired-end reads in mapping	$\mu_{\text{fragment}} + 3\sigma_{\text{fragment}}$
FRAG_MIN	Minimum fragment size from the paired-end reads in mapping	$\mu_{\text{fragment}} - 3\sigma_{\text{fragment}}$
Clone reconstruction parameters		
Parameter	**Explanation**	**Value**
WINDOW_SIZE	The minimum window size to look for potential clone seeds	μ_{fragment}
MIN_COVERAGE	The minimum coverage required for a window to be accepted as a clone seed	50-60%
EXTENSION	The distant from the edges of the clone seed to be extended to any fragment found, should be set to max fragment size	FRAG_MAX
Clone information for split clone discovery		
Parameter	**Explanation**	**Value**
CLONE_MEAN	The expected mean size of clones.	*from data*
CLONE_STD_DEV	The expected standard deviation of the clones.	*from data*
CLONE_MAX	The maximum possible clone length	$\mu_{\text{clone}} + 3\sigma_{\text{clone}}$
CLONE_MIN	The minimum possible clone length	$\mu_{\text{clone}} - 3\sigma_{\text{clone}}$

Continued on next page

Table 3.1 – *Continued from previous page (dipSeq parameters)*

Inversion information		
Parameter	**Explanation**	**Value**
INV_MIN_SIZE	Minimum inversion size to find	*user specific*
INV_MAX_SIZE	Maximum inversion size to find	*user specific*
INV_GAP	The distance between two split clones, should allow for one normal clone size	μ_{clone}
INV_OVERLAP	The overlap allowed for split clones, should be set according to maximum fragment size for smaller inversions and to the size of a clone for >500 Kbp	$-1\times$INV_GAP
INV_READ_LIMIT	The distance allowed around the split clones to find supporting reads, should allow for maximum fragment size	FRAG_MAX
Quasi-clique parameters		
Parameter	**Explanation**	**Value**
QCLIQUE_LAMBDA	The minimum percentage of k-clique nodes which should be present in the subgraph to be considered as a quasi-clique	0.5
QCLIQUE_GAMMA	The minimum percentage of k-clique edges which should be present in the subgraph to be considered as a quasi-clique	0.6
QCLIQUE_TABU	Number of rounds a node can be removed and added to a quasi-clique	$\log(\|V\|)$

Chapter 4

Testing and simulation

We designed three sets of simulation experiments to test and demonstrate the power of dipSeq for inversion discovery. In the first round we inserted simple inversions to test the correctness of dipSeq and optimize the parameters. The second simulation focuses on more realistic situations where inversion breakpoints spanned segmental duplications (SD). In the third simulation we investigated the effect of other SVs near or inside the breakpoints. Details of each experiment is given in the following sections. In all cases, chromosomes from the GRCh37 (hg19) was used. We tested both BAC and fosmid clones and mapped with BWA and mrFAST aligners.

4.1 Correctness and parameter tuning

In order to test the correctness of dipSeq, first, we randomly implanted 8 large inversions (500 Kbp to 10 Mbp) to the human reference genome (GRCh37) chromosome 1 (Table 4.1). Half of the simulated inversions were homozygous, and the remaining were heterozygous. We chose chromosome 1 because this is the biggest chromosome and, given that we must have avoid assembly gaps, allowed us to insert large inversions and later other structural variations (Section 4.3) with out overlapping. We then randomly selected BAC-sized intervals (μ = 150 Kbp, σ = 40 Kbp) from both chromosome 1 homologs at

Table 4.1: Inversions implanted on chromosome 1 for the simulation 1 and 3 experiments

ID	Start (bp)	End (bp)	Length (bp)	Genotype	SIM1	SIM3	Detectable
Inv1	4,676,939	6,950,520	2,273,580	Het (P)	4/2	0/3	Y/N
Inv2	69,598,859	72,079,080	2,480,220	Het (M)	2/3	10/6	Y/Y
Inv3	76,232,699	82,398,900	6,166,200	Hom	7/6	5+4/5+3	Y/Y
Inv4	94,844,699	98,902,620	4,057,920	Hom	8/5	3+4/5+2	Y/Y
Inv5	107,694,119	109,006,800	1,312,680	Het (P)	1/4	1/4	Y/Y
Inv6	171,527,459	176,658000	5,130,540	Het (M)	2/7	1/1	Y/Y
Inv7	185,266,199	187,919,700	2,653,500	Hom	11/5	2+3/3+2	Y/Y
Inv8	190,600,559	198,012,420	7,411,860	Hom	6/7	2+4/5+4	Y/Y

Genotype: Implanted inversions may be on one of the homologs (genotype=Het), or both (genotype=Hom). P: paternal, M: maternal copy.
SIM1: number of clones intersection the breakpoints in the first simulation (left/right)
SIM3: number of clones intersection the breakpoints in the third simulation (left/right)
Note that in the third simulation, in the homozygous inversions, since the SVs overlap or move the breakpoints they are no longer equivalent, thus two different number has been given (P+M).
Detectable: whether the inversion is detectable by dipSeq of not (simulation1/simulation3). dipSeq requires at least one clone to cover each breakpoint from different pools. Due to random cloning and low coverage (∼3X) sometimes the breakpoints would not be spanned by any clone.

∼3X physical coverage, which we randomly placed into 288 pools and simulated paired-end reads of length 100 bp (fragment size $\mu = 600$ bp, $\sigma = 60$ bp) using `wgsim`[1]. We generated three different data sets at 3X, 5X, and 10X depth of coverage to investigate the effect of read depth on our inversion calls. We mapped the reads to the reference genome using both BWA and mrFAST aligners and applied our clone reconstruction method. We were able to correctly infer 87.18% and 86.40% of the clones that were not located on the breakpoints using the BWA and mrFAST alignments, respectively (Table 4.2).

Using the inferred clones, dipSeq could find all 8 inversions at every coverage rate. It performed similarly in terms of sensitivity at all levels of depth of coverage, and returned no false positives. Table 4.7, Table 4.8, and Table 4.9 show the results obtained by dipSeq on the first simulation data using the BWA aligner at 3X, 5X, and 10X sequencing coverage, respectively, with the reads mapping on the same strand with a distance larger than the maximum fragment size ($\mu_{\text{fragment}} + 4\sigma_{\text{fragment}}$) used for paired end signals support. Table 4.10, Table 4.11, and Table 4.12 show the results obtained by dipSeq on the first

[1] `https://github.com/lh3/wgsim`

Table 4.2: Number of simulated clones correctly reconstructed by dipSeq with at least 90% reciprocal intersection

	P	M	P/M	percentage
Total Clones	5,079	5,001	10,080	100.00%
Inferred by BWA at 3X read depth	4,480	4,313	8,793	87.23%
Inferred by BWA at 5X read depth	4,478	4,309	8,787	87.17%
Inferred by BWA at 10X read depth	4,478	4,310	8,788	87.18%
Inferred by BWA at 15X read depth	4,477	4,311	8,788	87.18%
Inferred by BWA at 20X read depth	4,477	4,307	8,784	87.14%
Inferred by mrFAST at 3X read depth	4,448	4,255	8,703	86.34%
Inferred by mrFAST at 5X read depth	4,452	4,264	8,716	86.47%

P and M are the the paternal and maternal DNA, respectively.

simulation data for dipSeq using mrFAST aligner at 3X, 5X, and 10X sequencing coverage, respectively, with the alternative mappings marked as inversions in the DIVET file produced by the mrFAST aligner with edit distance <4 was used for the paired end signal support. Note that in this case the number of left and right support will be the same.

4.2 Robustness to segmental duplications

In the second experiment, we tested the robustness of dipSeq to segmental duplications, by implanting 4 large inversions (100 Kbp to 5 Mbp) to human chromosome 22, where the breakpoints intersect with segmental duplications (Table 4.3). We chose chromosome 22 because it is the smallest and the mapping would require less time. Two of the simulated inversions were homozygous, and the others were heterozygous. In addition, one of the inversions was placed near an assembly gap. We then randomly selected both BAC size (μ = 150 Kbp, σ = 40 Kbp) and fosmid size (μ = 40 Kbp, σ = 10 Kbp) intervals from both chromosome 22 homologs at ∼4X physical coverage, which we then randomly placed into 288 pools ensuring that the clones do not span the unmapped areas. We simulated paired-end reads of length 100 bp (fragment size μ = 600 bp, σ = 60 bp) using `wgsim` and generated three different data sets at 3X, 5X, and 10X depth of coverage, for both BAC and fosmid simulations.

Table 4.3: Inversions implanted on chromosome 22 with breakpoints placed on segmental duplications.

chromosome	**start locus**	**end locus**	**heterozygous or homozygous**
chr22	18,999,999	20,145,000	heterozygous (paternal)
chr22	22,606,699	29,075,000	homozygous
chr22	33,999,999	36,524,000	homozygous
chr22	42,105,089	44,963,000	heterozygous (maternal)

We mapped the reads to the entire reference genome using the BWA aligner, and finally applied dipSeq. Our algorithm was able to precisely detect all four inversions in each experiment, and returned no false positive predictions. We noticed that increasing the sequence coverage did not improve the results, but when the physical coverage was reduced to 3X, some inversions became undetectable since no clones spanned their breakpoints.

Table 4.13, Table 4.14, and Table 4.15 show the results obtained by dipSeq on the second simulation data using BAC clones and the BWA aligner at 3X, 5X, and 10X sequencing coverage, respectively, with the reads mapping on the same strand with a distance larger than the maximum fragment size ($\mu_{\text{fragment}} + 4\sigma_{\text{fragment}}$) used for paired end signals support. Table 4.16, Table 4.17, and Table 4.18 show the results obtained by dipSeq on the second simulation data using fosmid clones and the BWA aligner at 3X, 5X, and 10X sequencing coverage, respectively, with the reads mapping on the same strand with a distance larger than the maximum fragment size ($\mu_{\text{fragment}} + 4\sigma_{\text{fragment}}$) used for paired end signals support.

4.3 Robustness to the presence of other SVs

As a third simulation test, we explored dipSeq's performance when there are other forms of structural variation close to or intersecting the inversion breakpoints, therefore emulating complex rearrangements. We used the same simulated inversions of simulation 1 (Table 4.1), and we additionally implanted deletions and duplications (Table 4.4 and Table 4.5). We also inserted two additional inverted duplications to test whether dipSeq would predict them as normal inversions (Table 4.4). We then repeated our clone and paired-end read simulation (Section 4.1). However, due to random simulation, one of the

inversion breakpoints was not "detectable" i.e. no clones spanned the breakpoint (Table 4.1).

Table 4.19, Table 4.20, and Table 4.21 show the results obtained by dipSeq on the third simulation data using the BWA aligner at 3X, 5X, and 10X sequencing coverage, respectively, with the reads mapping on the same strand with a distance larger than the maximum fragment size ($\mu_{\text{fragment}} + 4\sigma_{\text{fragment}}$) used for paired end signals support. Table 4.22, Table 4.23, and Table 4.24 show the results obtained by dipSeq on the third simulation data for dipSeq using mrFAST aligner at 3X, 5X, and 10X sequencing coverage, respectively, with the alternative mappings marked as inversions in the DIVET file produced by the mrFAST aligner with edit distance <4 was used for the paired end signal support. Note that in this case the number of left and right support will be the same. All methods could retrieve the 7 discoverable inversions with no false positives except for mrFAST at 10X which suffered two false positive calls which shows that increasing the fragmenting coverage too high will not always result in better results. We have also shown that increasing the sequence coverage will worsen the clone reconstruction rate (Appendix A). In addition dipSeq did not identify inverted duplications as bona fide inversions.

Table 4.4: Duplications implanted on chromosome 1 for the third simulation

No.	Target Locus (Mbp)	Genotype (target)	Source Locus (Mbp)	Genotype (source)	Length (Mbp)	Site	Type
1	77	Hom	75-77	Hom	2	Inv3	Direct
2	81	Hom	83-84	Hom	2	Inv3	Direct
3	95	Het (P)	92-94	Het (M)	2	Inv4	Direct
4	97	Hom	98-99	Het (M)	1	Inv4	Direct
5	109	Hom	106.5-107.5	Het (M)	1	Inv5	Direct
6	174	Het (M)	175-177	Het (M)	2	Inv6	Direct
7	200	Hom	Inv7.start-Inv7.end	Hom	3	-	Inverted
8	221	Het (M)	217.8-219	Het (M)	1.2	-	Inverted
9*	223	Het (P)	217.8-219	Het (P)	1.2	-	Inverted

Duplications 1-6 were in direct orientation, and 7-9 were inverted.
Duplication #7 shares the same breakpoints with Inv7.
*The duplication was inserted twice.
Hom and Het are homozygous and heterozygous and P and M stand for paternal and maternal DNA.

Table 4.5: Deletions implanted on chromosome 1 for the third simulation

No.	Locus (Mbp)	Length (Mbp)	Genotype	Site
1	4.5-4.67	0.17	Hom	Inv1
2	4.68-4.7	0.02	Hom	Inv1
3	6.5-6.9	0.4	Het (P)	Inv1
4	7.0-7.6	0.6	Het (P)	Inv1
5	65-69.5	4.5	Het (M)	Inv2
6	72-73	1	Het (P)	Inv2

Deletions are simulated as either heterozygous or homozygous (genotype, P: paternal, M: maternal copy for heterozygous simulations).
Site: the ID of the closest implanted inversion (see Table 4.1).

4.4 Comparison to other tools

We further tested the efficacy of using whole genome sequencing (WGS) based inversion discovery algorithms on this data. For this purpose, we simulated WGS data sets, again using `wgsim`, at 3X, 5X, and 10X from the same chromosome homologs with the implanted inversions and SVs of simulation 3 (Table 4.1). We mapped the reads to the reference human genome (GRCh37) with both BWA and mrFAST, to test the detection performance of three algorithms: INVY [15], LUMPY [16], and VariationHunter [14]. We used the BWA alignments for INVY and LUMPY, and mrFAST alignments for VariationHunter, as per each tool's usage recommendations.

As expected, INVY and LUMPY failed to discover any of the implanted inversion events, as they are mainly designed for finding shorter inversions. VariationHunter was able to identify only one inversion out of 8, which may be due to VariationHunter's ability to incorporate all map locations, and a higher maximum inversion size threshold. The deletions it found are all incorrect.

Table 4.6: Results from VariationHunter on the simulation 3 data. At each coverage the same result was obtained.

Chr:chr1 Start:4,679,679-4,679,924 End:4,700,045-4,700,354 SVtype:D sup:3 Sum Weight:0 AvgEditDits:6.333333 LibSup:3 LibHurScore:3 AvgEditDistInd:6.33333 minDelLen:19,531 maxDelLen:20,331
Chr:chr1 Start:4,727,092-4,727,354 End:5,127,561-5,127,787 SVtype:D sup:6 Sum Weight:0 AvgEditDits:2.666667 LibSup:6 LibHurScore:6 AvgEditDistInd:2.66667 minDelLen:399,504 maxDelLen:400,304
Chr:chr1 Start:6,927,112-6,927,358 End:6,947,467-6,947,767 SVtype:D sup:4 Sum Weight:0 AvgEditDits:4.500000 LibSup:4 LibHurScore:4 AvgEditDistInd:4.5 minDelLen:19,555 maxDelLen:20355
Chr:chr1 Start:107,693,862-107,694,464 End:109,006,483-109,006,950 SVtype:V sup:5 Sum Weight:0 AvgEditDits:2.400000 LibSup:5 LibHurScore:5 AvgEditDistInd:2.4

Table 4.7: Simulation 1 results at 3X sequence coverage with the BWA aligner

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr1	4,676,834	4,676,941	6,950,341	6,950,523	4/10	9	4/10
chr1	69,598,666	69,598,985	72,078,771	72,079,641	11/7	24	11/7
chr1	76,232,635	76,232,701	82,398,750	82,398,912	8/13	24	8/13
chr1	94,844,639	94,844,699	98,902,086	98,902,652	5/14	27	5/14
chr1	107,694,087	107,694,177	109,006,650	109,006,857	1/4	6	1/4
chr1	171,527,266	171,527,459	176,657,976	176,658,043	11/9	20	11/9
chr1	185,266,111	185,266,201	187,919,391	187,920,258	4/11	21	4/11
chr1	190,600,382	190,600,561	198,012,231	198,012,420	10/11	24	10/11

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.8: Simulation 1 results at 5X sequence coverage with the BWA aligner

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr1	4,676,711	4,676,985	6,950,365	6,950,538	7/15	9	7/15
chr1	69,598,664	69,598,861	72,079,046	72,079,367	23/6	24	23/6
chr1	76,232,620	76,232,697	82,398,798	82,398,945	12/17	24	12/17
chr1	94,844,629	94,844,700	98,902,557	98,902,623	7/27	30	7/27
chr1	107,693,980	107,694,241	109,006,505	109,006,866	5/3	8	5/3
chr1	171,527,327	171,527,459	176,657,976	176,658,024	18/13	20	18/13
chr1	185,265,970	185,266,201	187,919,576	187,919,703	7/16	21	7/16
chr1	190,600,540	190,600,715	198,012,146	198,012,420	34/7	24	34/7

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.9: Simulation 1 results at 10X sequence coverage with the BWA aligner

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr1	4,676,780	4,676,941	6,950,466	6,950,521	24/28	9	24/28
chr1	69,598,822	69,598,861	72,078,996	72,079,083	50/21	24	50/21
chr1	76,232,586	76,232,701	82,398,805	82,398,903	43/50	24	42/50
chr1	94,844,576	94,844,700	98,902,553	98,902,623	19/67	30	19/67
chr1	107,694,058	107,694,121	109,006,701	109,006,835	16/7	8	16/7
chr1	171,527,415	171,527,459	176,657,931	176,658,002	31/32	20	31/32
chr1	185,266,045	185,266,200	187,919,633	187,919,702	15/46	21	15/46
chr1	190,600,465	190,600,561	198,012,259	198,012,420	53/18	24	53/18

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.10: Simulation 1 results at 3X sequence coverage using the alternative mappings given in the DIVET file obtained by the mrFAST aligner

chrom	left start	left end	right start	right end	PSC_{INV}	PSC	BP_{INV}
chr1	4,676,835	4,677,018	6,950,341	6,950,592	4/8	9	4
chr1	69,598,666	69,598,985	72,078,771	72,079,641	6/3	4	5
chr1	76,231,869	76,232,778	82,398,750	82,398,912	1/10	6	10
chr1	94,844,535	94,844,700	98,902,086	98,902,653	2/10	2	10
chr1	107,693,648	107,694,177	109,006,650	109,006,857	1/1	6	6
chr1	171,527,266	171,527,531	176,657,911	176,658,039	5/4	5	4
chr1	185,266,111	185,266,215	187,919,391	187,919,926	3/5	3	5
chr1	190,600,382	190,600,608	198,012,231	198,013,032	5/5	2	5

PSC_{INV}: number of inversions marked by INV in the DIVET file supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP_{INV}: number of inversions marked by INV in the DIVET file supporting the refined breakpoints.

Table 4.11: Simulation 1 results at 5X sequence coverage using the alternative mappings given in the DIVET file obtained by the mrFAST aligner

chrom	left start	left end	right start	right end	PSC_{INV}	PSC	BP_{INV}
chr1	4,676,711	4,676,985	6,950,365	6,950,538	4/8	9	4
chr1	69,598,664	69,598,883	72,078,822	72,079,367	14/3	14	3
chr1	76,232,554	76,232,694	82,398,798	82,399,020	6/13	6	13
chr1	94,844,346	94,844,710	98,902,404	98,902,651	5/17	5	17
chr1	107,693,980	107,694,241	109,006,586	109,006,866	9/2	9	2
chr1	171,527,327	171,527,502	176,657,816	176,658,127	9/5	9	5
chr1	185,265,970	185,266,210	187,919,576	187,919,739	4/11	4	11
chr1	190,600,257	190,600,715	198,012,146	198,012,435	6/5	6	5

PSC_{INV}: number of inversions marked by INV in the DIVET file supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP_{INV}: number of inversions marked by INV in the DIVET file supporting the refined breakpoints.

Table 4.12: Simulation 1 results at 10X sequence coverage using the alternative mappings given in the DIVET file obtained by the mrFAST aligner

chrom	left start	left end	right start	right end	PSC_{INV}	PSC	BP_{INV}
chr1	4,676,780	4,676,942	6,950,411	6,950,537	13/11	9	11
chr1	69,598,738	69,598,858	72,078,993	72,079,090	15/13	15	13
chr1	76,232,586	76,232,693	82,398,805	82,398,960	16/28	16	28
chr1	94,844,576	94,844,696	98,902,473	98,902,620	18/46	16	46
chr1	107,694,009	107,694,135	109,006,701	109,006,808	20/3	8	3
chr1	171,527,353	171,527,461	176,657,868	176,658,081	17/25	20	25
chr1	185,266,045	185,266,192	187,919,389	187,919,743	6/29	6	29
chr1	190,600,465	190,600,557	198,012,259	198,012,496	34/16	24	16

PSC_{INV}: number of inversions marked by INV in the DIVET file supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP_{INV}: number of inversions marked by INV in the DIVET file supporting the refined breakpoints.

Table 4.13: Simulation 2 results for BAC clones mapped with the BWA aligner at 3X sequence coverage

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr22	18,999,825	18,999,998	20,145,001	20,145,358	3/6	9	5/6
chr22	22,606,790	22,607,089	29,074,917	29,075,100	15/6	22	20/ 6
chr22	33,999,534	34,000,000	36,523,854	36,524,146	1/10	20	13/10
chr22	42,105,031	42,105,090	44,962,358	44,963,003	7/4	9	7/4

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.14: Simulation 2 results for BAC clones mapped with the BWA aligner at 5X sequence coverage

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr22	18,999,747	19,000,000	20,144,833	20,145,367	8/4	9	10/4
chr22	22,606,888	22,607,068	29,074,930	29,075,002	23/12	22	23/12
chr22	33,999,937	34,000,000	36,523,984	36,524,017	15/19	20	15/19
chr22	42,104,773	42,105,090	44,963,001	44,963,112	7/7	9	10/4

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.15: Simulation 2 results for BAC clones mapped with the BWA aligner at 10X sequence coverage

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr22	19,000,000	19,000,000	20,145,002	20,145,002	15/8	9	15/8
chr22	22,606,979	22,607,000	29,075,003	29,075,001	43/17	22	43/17
chr22	33,999,971	34,000,028	36,523,951	36,524,002	30/26	20	30/26
chr22	42,105,090	42,105,140	44,963,001	44,963,002	24/17	9	15/8

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.16: Simulation 2 results for fosmid clones mapped with the BWA aligner at 3X sequence coverage

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr22	18,999,962	19,000,004	20,144,782	20,145,002	6/3	6	6/3
chr22	22,606,369	22,607,000	29,074,592	29,075,584	1/11	20	6/11
chr22	33,999,762	34,000,000	36,523,799	36,524,002	8/7	12	10/7
chr22	42,105,006	42,105,246	44,962,974	44,963,094	2/6	1	2/6

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.17: Simulation 2 results for fosmid clones mapped with the BWA aligner at 5X sequence coverage

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr22	18,999,775	19,000,000	20,145,001	20,145,002	6/6	6	6/6
chr22	22,607,000	22,607,000	29,075,003	29,075,093	17/15	15	17/15
chr22	33,999,946	34,000,000	36,523,649	36,524,024	15/18	20	15/18
chr22	42,105,090	42,105,308	44,963,001	44,963,004	3/9	2	3/11

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.18: Simulation 2 results for fosmid clones mapped with the BWA aligner at 10X sequence coverage

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr22	18,999,910	19,000,000	20,145,001	20,145,002	15/19	6	15/19
chr22	22,607,000	22,606,998	29,075,003	29,075,004	38/21	20	38/21
chr22	33,999,954	34,000,000	36,524,001	36,524,002	23/22	20	23/22
chr22	42,105,009	42,105,090	44,962,824	44,963,002	13/19	2	13/19

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.19: Simulation 3 results for BWA aligner at 3X sequence coverage

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr1	69,598,700	69,598,861	72,078,819	72,079,086	12/8	3	12/8
chr1	76,232,671	76,232,701	82,398,281	82,398,903	33/7	13	33/7
chr1	94,844,615	94,844,700	98,902,491	98,902,886	13/18	6	13/18
chr1	107,499,795	107,568,153	108,891,152	108,979,319	2/1	1	0/0
chr1	171,527,333	171,527,459	176,657,966	176,658,003	5/3	4	5/3
chr1	185,266,097	185,266,226	187,919,308	187,919,755	7/8	4	7/8
chr1	190,600,405	190,600,561	198,012,320	198,012,772	15/6	7	15/6

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.20: Simulation 3 results for BWA aligner at 5X sequence coverage

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr1	69,598,710	69,598,860	72,078,766	72,079,083	26/14	4	26/14
chr1	76,232,516	76,232,698	82,398,843	82,398,943	48/11	13	48/11
chr1	94,844,540	94,844,700	98,902,374	98,902,816	22/40	6	22/40
chr1	107,693,988	107,694,121	109,006,493	109,006,998	1/6	1	1/6
chr1	171,527,312	171,527,458	176,657,887	176,658,099	9/15	1	9/15
chr1	185,266,150	185,266,201	187,919,652	187,919,706	12/10	4	12/10
chr1	190,600,428	190,600,561	198,012,352	198,012,420	25/14	7	25/14

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.21: Simulation 3 results for BWA aligner at 10X sequence coverage

chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
chr1	69,598,788	69,598,861	72,078,864	72,079,083	45/27	4	45/27
chr1	76,232,516	76,232,696	82,398,875	82,398,903	67/46	13	67/46
chr1	94,844,475	94,844,700	98,902,491	98,902,623	37/79	6	37/79
chr1	107,694,061	107,694,121	109,006,687	109,006,803	4/15	4	4/15
chr1	171,527,415	171,527,459	176,657,801	176,658,003	23/37	1	23/37
chr1	185,266,101	185,266,201	187,919,641	187,919,703	34/18	4	34/18
chr1	190,600,270	190,600,561	198,012,253	198,012,420	26/28	7	26/28

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 4.22: Simulation 3 results for mrFAST aligner at 3X sequence coverage using alternative mappings given in the DIVET file

chrom	left start	left end	right start	right end	PSC_{INV}	PSC	BP_{INV}
chr1	69,598,538	69,598,883	72,078,971	72,079,224	4/2	4	2
chr1	76,232,548	76,232,843	82,398,281	82,399,081	13/3	14	3
chr1	94,844,416	94,844,701	98,902,491	98,902,768	7/10	9	10
chr1	107,693,958	107,694,370	109,006,676	109,006,917	1/4	4	4
chr1	171,527,333	171,527,501	176,657,804	176,658,190	4/3	1	3
chr1	185,266,097	185,266,226	187,919,308	187,919,755	4/4	6	4
chr1	190,600,405	190,600,565	198,012,320	198,012,473	7/4	9	4

PSC_{INV}: number of inversions marked by INV in the DIVET file supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP_{INV}: number of inversions marked by INV in the DIVET file supporting the refined breakpoints.

Table 4.23: Simulation 3 results for mrFAST aligner at 5X sequence coverage using alternative mappings given in the DIVET file

chrom	left start	left end	right start	right end	PSC_{INV}	PSC	BP_{INV}
chr1	69,598,710	69,598,861	72,078,767	72,079,099	10/10	10	10
chr1	76,232,516	76,232,698	82,398,623	82,398,943	22/6	22	6
chr1	94,844,540	94,844,733	98,902,374	98,902,620	12/28	12	28
chr1	107,693,989	107,694,214	109,006,493	109,006,998	1/6	4	6
chr1	171,527,312	171,527,459	176,657,887	176,658,100	7/7	1	7
chr1	185,266,067	185,266,195	187,919,478	187,919,793	4/5	4	5
chr1	190,600,428	190,600,557	198,012,265	198,012,598	9/7	9	7

PSC_{INV}: number of inversions marked by INV in the DIVET file supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP_{INV}: number of inversions marked by INV in the DIVET file supporting the refined breakpoints.

Table 4.24: Simulation 3 results for mrFAST aligner and 10X coverage using alternative mappings given in the DIVET file.

chrom	left start	left end	right start	right end	PSC_{INV}	PSC	BP_{INV}
chr1	69,598,738	69,598,865	72,078,911	72,079,129	13/17	12	17
chr1	76,232,589	76,232,695	82,398,757	82,398,924	12/21	12	21
chr1	94,844,589	94,844,705	98,902,486	98,902,657	14/14	14	14
chr1	107,694,017	107,694,118	109,006,506	109,006,846	7/1	2	1
chr1	145,333,689	145,342,656	148,329,447	148,321,572	2/1	9	0
chr1	145,333,653	145,342,742	148,015,682	148,011,520	2/7	1	1
chr1	171,527,320	171,527,463	176,657,906	176,658,084	9/7	1	7
chr1	185,266,027	185,266,199	187,919,523	187,919,764	6/26	4	5
chr1	190,600,446	190,600,557	198,012,281	198,012,420	26/7	14	7

PSC_{INV}: number of inversions marked by INV in the DIVET file supporting the paired split clones via simple summation. PSC: number of paired split clones supporting the breakpoints. BP_{INV}: number of inversions marked by INV in the DIVET file supporting the refined breakpoints.

Chapter 5

Experimental results

After proving the correctness and robustness of dipSeq on simulated data, we tested it on the real data of the NA12878 individual.

5.1 Building pooled clone libraries

First a single whole-genome BAC library with long inserts (∼140 Kbp) was produced by Joyce Tang and Chris T. Amemiya at the Benaroya Research Institute, United States. This procedure is a modification of the original haplotyping method previously described by Kitzman et al. (2011), that generates fosmid libraries with ∼40 Kbp inserts. Here we use BAC clones, since long inserts are required to span the large duplication blocks where inversion breakpoints typically map [20, 29]. We then randomly partition the library into pools such that each pool is essentially a haploid mixture of clones derived from either the maternal or paternal DNA at each genomic location. High-throughput sequencing of each pool provides haplotype information for each clone in that pool.

We used genomic DNA from a HapMap Project individual (NA12878) to construct the

BAC library. High molecular weight DNA was isolated, partially EcoRI digested, and subcloned into pCC1BAC vector (Epicentre) to create a ∼140 Kbp insert library using previously described protocols [65]. We then split a portion of this library to 3 sets of 96 pools each, with 230 clones per pool for set 1, 389 clones per pool for set 2 and 153 clones per pool for set 3. Each pool was expanded by direct liquid outgrowth after infection. We next construct 96 barcoded sequencing libraries per each set, for a total of 288 sequencing libraries [66]. Libraries from each set were indexed with barcodes, combined and sequenced using the Illumina HiSeq platform (101 bp paired-end reads). Upon sequencing a total of 74,112 clones (22,080 in Set 1, 37,344 in Set 2 and 14,688 in Set 3) we obtained 3.38X expected physical depth of coverage. After read mapping and clone reconstruction (Section 3.3.1), 87.58% of the genome was covered by one or more clones. This part of the pooled clone sequencing was done by Mattia Miroballo and Francesca Antonacci at the Department of Biology, University of Bari, Italy.

5.2 Inversions predicted on the real dataset from NA12878

Next, we tested dipSeq using a real pooled clone sequencing dataset generated from the genome of NA12878. We mapped the paired-end reads from a total of 288 pools using both BWA and mrFAST to the reference genome. Average fragment length of the paired-end reads was ∼450 bp, with a standard deviation of ∼98 bp. Using our algorithms, we reconstructed the clone locations, which showed an average clone length of ∼140 Kbp and a standard deviation of 40 Kbp.

Figure 5.1(A) shows the clone size histogram for all sets where Figure 5.1(B), Figure 5.1(C), and Figure 5.1(D) show the clone size histogram on each set separately. As it can be observed set 3 shows very poor quality and there are too many split clones in this set. When mapping the reads we noticed that many pools were empty or contaminated with bacteria genome and thus the sequencing coverage fell to low to reconstruct the clones.

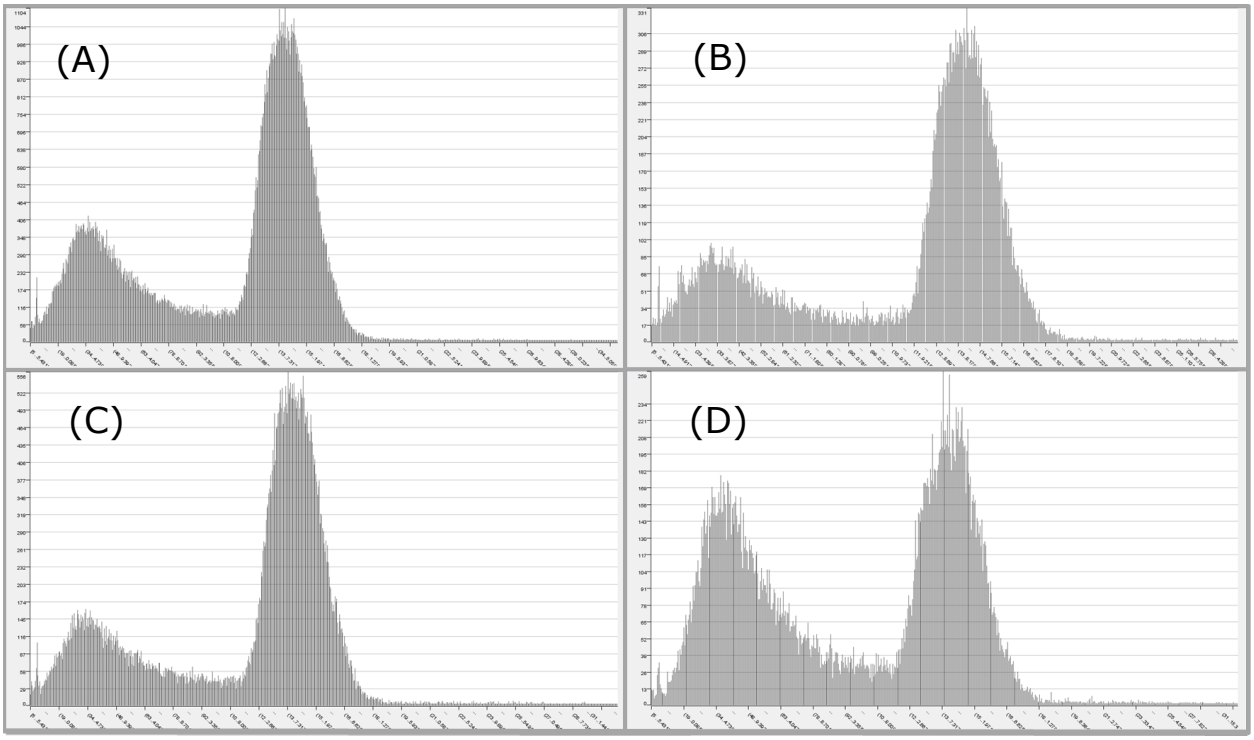


Figure 5.1: Inferred clone size histogram for each by set

X: clone size Y: number of clones
(A) all sets (B) set 1 (C) set 2 (D) set 3

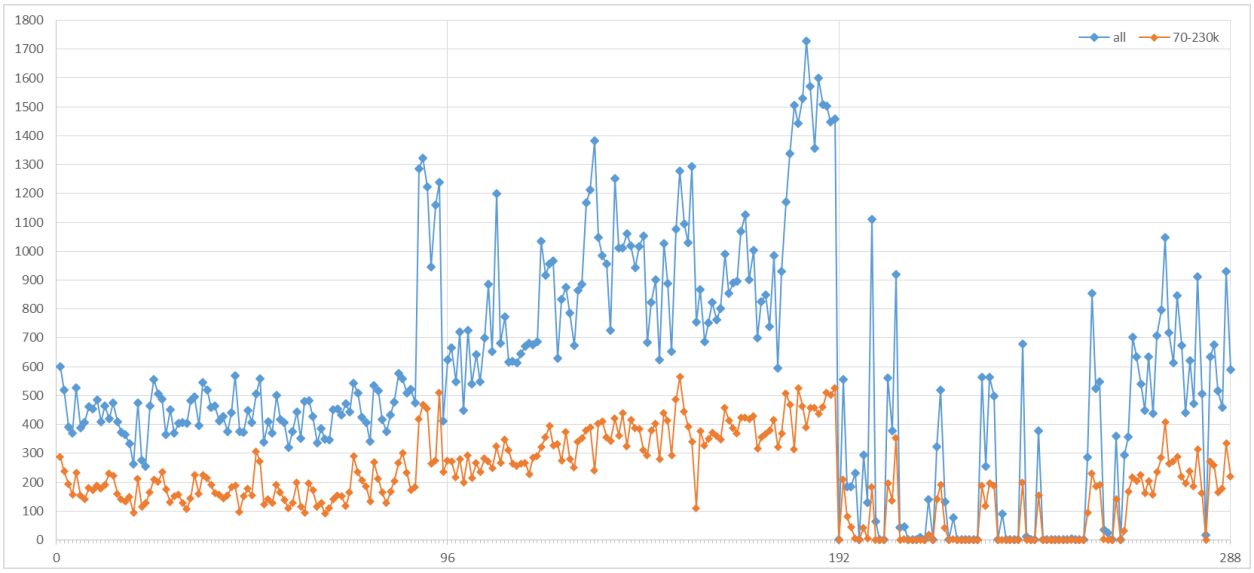


Figure 5.2: Number of inferred clones in each pool.

X: pool number Y: number of clones

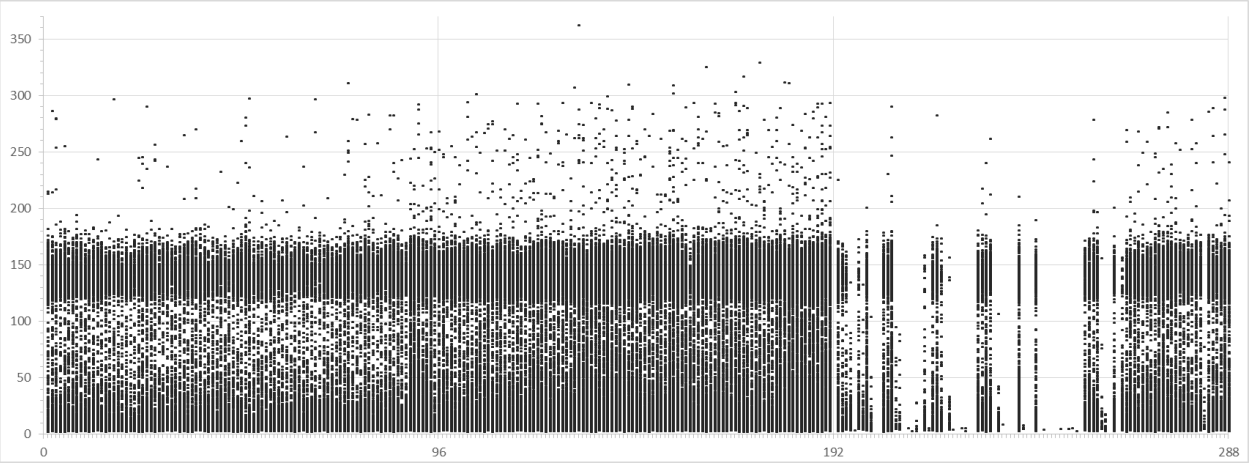


Figure 5.3: Size of clones in each pool.
X: pool number Y: size of clones (Kbp)

Figure 5.2 illustrates the number of clones found in each pool where pools 1–96 represent set 1, 97–192 represent set 2 and 193–288 represent set 3. The blue line shows the total number of clones which is not very informative. The orange line shows the number of normal clones in range [$\mu_{\text{clone}} - \sigma_{\text{clone}}$=70 Kbp, $\mu_{\text{clone}} + \sigma_{\text{clone}}$=230 Kbp] which corresponds to the expected number of clones in each set as mentioned above (230 clones per pool for set 1, 389 clones per pool for set 2, and 153 clones per pool for set 3).

Figure 5.3 shows a scatter plot of clone size in each pool. Again we can see that in set 3 we have very poor coverage and most pools are completely empty. In addition it can be observed that as the number of clones per pool increase more large clones will be reconstructed which is a result of overlapping clones. For example pools of set 2 show much more large clone size outliers than pools of set 1.

For inversion discovery, we set the minimum and maximum inversion size thresholds as 100 Kbp and 10 Mbp, respectively. Although it is theoretically possible to detect inversions as small as a typical clone size (150-200 Kbp), due to the limitations of the FISH method, we cannot validate inversions <500 Kbp. We generated two main callsets using BWA and mrFAST, where >83% of the calls were shared as follows:

On the pooled BAC clones from the NA12878 genome, we applied dipSeq using two aligners with two different parameter sets. In the first method (BW), we aligned the

paired-end reads using BWA and separated the paired-end reads that map in the same orientation for the support calculation. In the second method (MF), we used the mrFAST aligner and retrieved the DIVET file provided by mrFAST which marks all the alternative mappings of the paired-end reads along with the type of signature they produce. This file is supposed to be used with Variation Hunter, which simply applies a set cover approximation on these alternative mappings. Here we use these potential inversions in the DIVET file, but this time we do not look for support on the left and right breakpoints as before, but instead we count the number of marked inversions that overlap with the inversion breakpoints we have found. Observing that the DIVET file contains too many alternative mappings, we trimmed the alternatives with edit distance ≤ 4. Then using the two sets of data, we aimed to detect inversions of size 500 Kbp–10 Mbp, and in another run 100 Kbp–500 Kbp, thus setting the min and max inversion sizes for each run accordingly. The reason of separating the min and max inversion set was the limitation of FISH experiments. After running dipSeq and obtaining the clusters, we removed clusters with 0 support on any breakpoint. Then, for each set of inversions using bedtools [67] we removed the inversions that overlap with the known gaps with at least 1% intersection.

Table 5.1 gives the predicted breakpoints on the data of the NA12878 individual using the BWA aligner and setting the minimum and maximum inversion parameters to 100 Kbp and 500 Kbp. Table 5.2 gives the predicted breakpoints on the data of the NA12878 individual using the BWA aligner and setting the minimum and maximum inversion parameters to 500 Kbp and 10 Mbp. Table 5.3 gives the predicted breakpoints on the data of the NA12878 individual using the mrFAST aligner and setting the minimum and maximum inversion parameters to 100 Kbp and 500 Kbp. Table 5.4 gives the predicted breakpoints on the data of the NA12878 individual using the mrFAST aligner and setting the minimum and maximum inversion parameters to 500 Kbp and 10 Mbp. In the case that the BWA aligner was used to align the reads, the reads mapped to the same strand with a distance larger than expected ($> \mu_{\text{fragment}} + 4\sigma_{\text{fragment}}$) were used for calculating the paired read signal support. And in the case that mrFAST was used to aligned the reads, the alternative mappings given in the DIVET file which were marked as inversion with edit distance <4 were used for calculating the paired read signal support. Note that in this case the left and right support will be the same.

For the sake of readability, we have assigned unique IDs to the inversions detected by

dipSeq, where the prefix BW is used for the BWA aligner and MF for mrFAST.

Table 5.1: Inversions of size 100–500 Kbp predicted on NA12878 individual using the BWA aligner.

ID	chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
BW1	chr2	87,276,424	87,263,205	87,384,777	87,377,676	10/7	1	16/13
BW2	chr2	91,992,806	91,985,955	92,252,738	92,234,019	6/8	2	2/9
BW3	chr7	64,635,234	64,624,518	64,998,550	64,986,359	2/3	1	2/3
BW4	chr10	42,401,131	42,382,707	42,535,677	42,527,092	981/1,299	5	1,029/1,299
BW5	chr11	50,096,398	50,077,604	50,324,272	50,321,339	10/1	1	3/3
BW6	chr13	52,809,864	52,729,423	53,152,092	53,159,691	12/6	2	17/14
BW7	chr16	14,864,601	14,849,638	15,457,302	15,440,576	1/5	2	2/3
BW8	chr16	21,450,227	21,456,067	21,883,232	21,866,065	6/3	2	1/5
BW9	chr16	69,992,135	69,978,097	70,208,270	70,221,477	100/24	8	92/66
BW10	chr17	44,391,797	44,385,946	44,609,394	44,604,209	2/31	2	2/5
BW11	chrX	140,114,110	140,104,877	140,659,531	140,656,893	1/2	1	1/3

AB/CD: the paired read support ($++/--$) reads supporting the paired split clones via simple summation.

PSC: number of paired split clones supporting the breakpoints.

BP1/BP2: number of paired read support ($++/--$) reads that supports the refined breakpoints.

Table 5.2: Inversions of size 500 Kbp –10 Mbp predicted on NA12878 individual using the BWA aligner

ID	chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
BW12	chr2	96,642,563	96,592,119	97,798,756	97,824,376	2/5	8	24/24
BW13	chr2	107,027,986	107,055,302	108,534,871	108,446,654	9/4	7	5/4
BW14	chr2	130,746,185	130,794,939	132,129,381	132,046,989	1/8	4	29/19
BW15	chr3	123,716,705	123,667,775	125,690,261	125,703,585	2/4	1	9/13
BW16	chr5	69,006,219	68,925,445	70,086,940	70,015,749	29/25	40	29/26
BW17	chr5	175,656,185	175,692,620	177,170,160	177,055,602	1/2	1	4/2
BW18	chr7	32,869,843	32,868,354	35,028,527	34,945,706	1/1	6	1/1
BW19	chr7	51,501,166	51,457,375	56,451,750	56,461,453	10/8	2	37/12

Continued on next page

Table 5.2 – *Continued from previous page (BWA 500 Kbp –10 Mbp)*

ID	chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
BW20	chr7	66,676,869	66,598,823	72,104,443	72,110,027	3/8	6	5/9
BW21	chr7	72,558,496	72,593,223	74,358,790	74,488,912	15/15	15	16/27
BW22	chr8	6,966,003	6,964,085	12,574,044	12,496,761	10/10	1	20/14
BW23	chr9	86,505,260	86,441,747	88,382,909	88,454,500	1/2	4	7/14
BW24	chr9	97,177,469	97,062,107	99,672,076	99,780,119	6/4	22	18/11
BW25	chr10	81,501,494	81,460,982	89,005,838	89,036,463	4/1	6	11/10
BW26	chr14	19,400,241	19,387,370	20,165,005	20,158,565	6/13	1	12/11
BW27	chr14	19,426,017	19,427,037	20,134,688	20,129,457	2/4	6	4/5
BW28	chr14	19,607,629	19,597,366	19,976,139	19,979,958	7/1	3	5/8
BW29	chr15	22,811,076	22,729,323	28,662,344	28,793,101	2/5	4	8/6
BW30	chr15	23,124,573	23,094,916	29,076,834	29,089,344	11/2	1	10/7
BW31	chr15	30,728,410	30,819,779	32,850,269	32,761,686	5/6	49	122/137
BW32	chr15	74,363,275	74,360,818	75,595,294	75,561,305	1/5	1	1/1
BW33	chr15	83,000,584	83,009,461	84,958,581	84,935,456	3/1	3	1/1
BW34	chr15	100,322,230	100,348,404	102,336,630	102,212,197	1/1	1	3/3
BW35	chr16	14,998,481	14,889,883	18,511,593	18,574,072	1/7	2	8/11
BW36	chr16	15,341,343	15,275,762	16,609,155	16,611,192	1/1	1	1/1
BW37	chr16	16,742,663	16,655,533	18,743,451	18,762,657	1/2	2	6/4
BW38	chr16	21,450,984	21,417,395	22,515,005	22,505,491	6/7	3	9/4
BW39	chr16	21,893,378	21,841,285	29,390,804	29,535,555	6/8	30	35/36
BW40	chr16	22,545,044	22,511,008	30,281,972	30,200,002	13/12	17	13/13
BW41	chr16	70,233,551	70,221,140	74,483,623	74,424,445	44/3	1	41/10
BW42	chr17	15,619,886	15,474,088	18,569,007	18,581,660	1/1	1	4/2
BW43	chr17	18,471,322	18,486,307	20,287,002	20,247,505	1/1	2	11/6
BW44	chr17	34,823,474	34,739,949	36,294,232	36,250,175	50/55	4	50/54
BW45	chr17	58,293,792	58,286,999	60,371,164	60,255,941	2/3	2	3/3
BW46	chr18	10,728,143	10,632,783	12,180,972	12,176,363	2/3	2	2/6

AB/CD: the paired read support (++/−−) reads supporting the paired split clones via simple summation.

PSC: number of paired split clones supporting the breakpoints.

BP1/BP2: number of paired read support (++/−−) reads that supports the refined breakpoints.

Table 5.3: Inversions of size 100–500 Kbp predicted on NA12878 individual using the DIVET file given by the mrFAST aligner with edit distance ≤ 4.

ID	chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
MF1	chr1	144,829,408	144,828,037	145,370,642	145,368,122	3/13	10	9
MF2	chr2	87,276,424	87,263,205	87,384,777	87,377,676	24/22	1	32
MF3	chr2	91,992,806	91,985,955	92,252,738	92,234,019	59/68	2	58
MF4	chr2	130,812,780	130,824,289	131,206,878	131,214,049	10/23	2	58
MF5	chr7	64,635,234	64,624,518	64,998,550	64,986,359	7/7	1	7
MF6	chr7	64,973,294	64,971,084	65,115,077	65,106,150	2/2	10	2
MF7	chr7	143,907,123	143,895,393	144,046,137	144,047,619	16/2	1	14
MF8	chr9	46,700,437	46,696,377	46,834,953	46,823,887	16/24	1	19
MF9	chr10	42,401,131	42,382,707	42,535,677	42,527,092	16/29	5	30
MF10	chr16	14,877,988	14,877,924	15,441,797	15,432,606	8/12	1	10
MF11	chr16	21,450,227	21,456,067	21,883,232	21,866,065	174/18	3	88
MF12	chr16	21,500,476	21,482,090	21,931,837	21,913,853	573/91	2	49
MF13	chr16	69,992,135	69,978,097	70,208,270	70,221,477	77/8	8	89
MF14	chr17	44,369,469	44,369,719	44,586,239	44,586,382	35/12	18	2
MF15	chr17	44,391,797	44,400,349	44,609,394	44,618,085	12/7	1	12

AB/CD: the paired read support $(++/--)$ reads supporting the paired split clones via simple summation.

PSC: number of paired split clones supporting the breakpoints.

BP1/BP2: number of paired read support $(++/--)$ reads that supports the refined breakpoints.

Table 5.4: Inversions of size 500 Kbp–10 Mbp predicted on NA12878 individual using the DIVET file given by the mrFAST aligner with edit distance ≤ 4.

ID	chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
MF16	chr1	146,395,694	146,467,444	147,615,065	147,551,865	9/6	3	68
MF17	chr2	96,625,361	96,499,871	97,833,880	97,878,905	9 /25	1	112
MF18	chr2	107,080,187	107,067,229	108,534,871	108,446,238	33/14	2	14
MF19	chr2	111,327,858	111,280,588	113,186,168	113,072,509	22/15	1	16
MF20	chr2	130,812,780	130,826,487	132,112,334	132,023,394	49/3	10	69
MF21	chr3	123,716,705	123,667,775	125,690,261	125,703,585	2/4	1	18
MF22	chr5	21,522,888	21,508,067	29,438,065	29,447,267	1/16	1	22
MF23	chr5	68,933,817	68,951,654	70,421,229	70,540,747	5/4	310	644

Continued on next page

Table 5.4 – *Continued from previous page (mrFAST 500 Kbp–10 Mbp)*

ID	chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
MF24	chr5	175,656,185	175,692,620	177,170,160	177,055,602	2/2	4	13
MF25	chr7	32,869,843	32,868,354	35,028,527	34,945,706	3/3	6	3
MF26	chr7	51,501,166	51,457,375	56,451,750	56,461,453	8/5	2	22
MF27	chr7	66,770,491	66,641,494	72,027,054	72,053,018	2/2	15	12
MF28	chr7	72,558,496	72,593,223	74,358,790	74,488,912	116/8	17	164
MF29	chr8	7,051,767	7,020,318	12,480,229	12,483,867	63/10	9	51
MF30	chr9	86,505,260	86,441,747	88,382,909	88,454,500	1/5	4	15
MF31	chr9	97,177,469	97,062,107	99,672,076	99,715,862	4/1	51	10
MF32	chr10	81,501,494	81,460,982	89,005,838	89,019,684	4/5	5	13
MF33	chr14	19,426,017	19,427,037	20,134,688	20,129,457	2/7	4	5
MF34	chr14	63,570,919	63,584,607	67,140,192	67,042,279	1/32	2	17
MF35	chr15	23,495,679	23,391,111	28,625,879	28,761,245	132/12	41	48
MF36	chr15	30,728,410	30,825,273	32,850,269	32,761,686	77/17	74	110
MF37	chr15	74,363,275	74,360,818	75,595,294	75,561,305	2/12	1	3
MF38	chr15	82,931,688	83,009,461	84,835,661	84,935,456	2/1	11	20
MF39	chr15	100,322,230	100,348,404	102,336,630	102,212,197	3/1	1	15
MF40	chr16	12,026,273	12,021,483	21,893,378	21,843891	2/1	3	2
MF41	chr16	14,938,250	15,029,573	16,324,009	16,422,753	10/52	405	84
MF42	chr16	15,205,940	15,249,321	18,418,056	18,490,186	10/6	20	117
MF43	chr16	16,791,146	16,649,631	18,715,257	18,685,688	13/12	3	13
MF44	chr16	18,868,779	18,933,343	21,512,037	21,432,448	1/2	2	48
MF45	chr16	21,434,903	21,455,882	30,336,523	30,206,538	49/28	6	447
MF46	chr16	21,765,809	21,758,040	22,575,584	22,574,102	9/9	3	6
MF47	chr16	21,893,378	21,841,285	29,532,007	29,535,555	246/151	26	224
MF48	chr16	22,530,124	22,509,013	28,651,702	28,670,877	16/5	1	24
MF49	chr16	31,987,421	31,911,754	33,750,338	33,764,390	5/2	6	15
MF50	chr16	32,351,324	32,328,258	33,354,611	33,379,404	2/6	8	2
MF51	chr16	70,233,551	70,221,140	74,483,623	74,424,445	10/7	1	7
MF52	chr17	15,619,886	15,474,088	18,569,007	18,587,824	2/5	25	14
MF53	chr17	18,454,611	18,479,530	20,310,049	20,247,505	4/8	7	19
MF54	chr17	36,350,387	36,226,733	45,524,595	45,615,353	5/4	39	29
MF55	chr17	43,591,979	43,605,052	45,153,801	45,107,620	3/3	2	12

Continued on next page

Table 5.4 – *Continued from previous page (mrFAST 500 Kbp–10 Mbp)*

ID	chrom	left start	left end	right start	right end	AB/CD	PSC	BP1/BP2
MF56	chr17	58,284,564	58,278,096	60,381,831	60,323,082	6/10	2	7
MF57	chr18	5,354,256	5,329,111	13,978,445	14,105,010	7/1	1	5
MF58	chr18	10,649,481	10,585,261	12,201,977	12,209,526	3/13	2	7

AB/CD: the paired read support (++/−−) reads supporting the paired split clones via simple summation.
PSC: number of paired split clones supporting the breakpoints.
BP1/BP2: number of paired read support (++/−−) reads that supports the refined breakpoints.

5.3 BWA, mrFAST and InvFEST compared

Furthermore, the predicted inversions by dipSeq with the BWA and mrFAST aligner are compared against each other and the inversions given in the InvFEST database [41]. In Table 5.5 the inversions predicted using the BWA aligner is compared to others. In Table 5.6 the inversions predicted using the mrFAST aligner is compared to others. In Table 5.7 the inversions published in the InvFEST on the NA12878 individual with size>90,000 that could be lifted over using the UCSC `liftOver` tool [1] were extracted and compared to the dipSeq results. In each table if the inversion breakpoints were validated using FISH experiments (see Section 5.4) the result is given in the last row.

Table 5.5: BWA inversions compared against mrFAST inversions, InvFEST and the callset

BW ID	**chrom**	**inner start**	**inner end**	**size**	**MF ID**	**InvFEST**	**callset result**
BW1	chr2	87,263,205	87,384,777	121,572	MF2	HsInv0242	
BW2	chr2	91,985,955	92,252,738	266,783	MF3		
BW3	chr7	64,624,518	64,998,550	374,032	MF5	HsInv0484	
BW4	chr10	42,382,707	42,535,677	152,970	MF9		
BW5	chr11	50,077,604	50,324,272	246,668		HsInv0330	
BW6	chr13	52,729,423	53,152,092	422,669		HsInv0759	
BW7	chr16	14,849,638	15,457,302	607,664	MF10	HsInv0551	
BW8	chr16	21,456,067	21,883,232	427,165	MF11		
BW9	chr16	69,978,097	70,208,270	230,173	MF13		

Continued on next page

[1] https://genome.ucsc.edu/cgi-bin/hgLiftOver

Table 5.5 – *Continued from previous page (BWA vs mrFAST)*

BW ID	chrom	inner start	inner end	size	MF ID	InvFEST	callset result
BW10	chr17	44,385,946	44,609,394	223,448	MF15		
BW11	chrX	140,104,877	140,659,531	554,654			
BW12	chr2	96,592,119	97,798,756	1,206,637	MF17		
BW13	chr2	107,055,302	108,534,871	1,479,569	MF18		
BW14	chr2	130,794,939	132,129,381	1,334,442	MF20	HsInv0669 HsInv0698	
BW15	chr3	123,667,775	125,690,261	2,022,486	MF21		
BW16	chr5	68,925,445	70,086,940	1,161,495	MF23		*not tested*
BW17	chr5	175,692,620	177,170,160	1,477,540	MF24	HsInv0687 HsInv0273 HsInv0281 HsInv0280 HsInv0276	
BW18	chr7	32,868,354	35,028,527	2,160,173	MF25		
BW19	chr7	51,457,375	56,451,750	4,994,375	MF26		
BW20	chr7	66,598,823	72,104,443	5,505,620	MF27		
BW21	chr7	72,593,223	74,358,790	1,765,567	MF28		
BW22	**chr8**	**6,964,085**	**12,574,044**	**5,609,959**	**MF29**	**HsInv0501** **HsInv0496** **HsInv0713** **HsInv0717** **HsInv0494** **HsInv0497**	**confirmed**
BW23	chr9	86,441,747	88,382,909	1,941,162	MF30		
BW24	chr9	97,062,107	99,672,076	2,609,969	MF31		
BW25	chr10	81,460,982	89,005,838	7,544,856	MF32		
BW26	chr14	19,387,370	20,165,005	777,635	MF33		
BW27	chr14	19,427,037	20,134,688	707,651	MF33		
BW28	chr14	19,597,366	19,976,139	378,773			
BW29	chr15	22,729,323	28,662,344	5,933,021	MF35		*not confirmed*
BW30	chr15	23,094,916	29,076,834	5,981,918			
BW31	**chr15**	**30,819,779**	**32,850,269**	**2,030,490**	**MF36**	**HsInv1049**	**confirmed**

Continued on next page

Table 5.5 – *Continued from previous page (BWA vs mrFAST)*

BW ID	chrom	inner start	inner end	size	MF ID	InvFEST	callset result
BW32	chr15	74,360,818	75,595,294	1,234,476	MF37	HsInv0544 HsInv0771	
BW33	**chr15**	**83,009,461**	**84,958,581**	**1,949,120**	**MF38**	**HsInv0547**	***confirmed***
BW34	chr15	100,348,404	102,336,630	1,988,226	MF39		
BW35	chr16	14,889,883	18,511,593	3,621,710	MF42		
BW36	chr16	15,275,762	16,609,155	1,333,393	MF41		
BW37	chr16	16,655,533	18,743,451	2,087,918	MF43		
BW38	chr16	21,417,395	22,515,005	1,097,610	MF46	HsInv0152	
BW39	chr16	21,841,285	29,390,804	7,549,519	MF47		*not confirmed*
BW40	chr16	22,511,008	30,281,972	7,770,964			
BW41	chr16	70,199,886	74,452,605	4,252,719	MF51		*not confirmed*
BW42	chr17	15,474,088	18,569,007	3,094,919	MF52	HsInv0373	
BW43	chr17	18,486,307	20,287,002	1,800,695	MF53		
BW44	**chr17**	**34,739,949**	**36,294,232**	**1,554,283**		**HsInv1048**	***confirmed***
BW45	chr17	58,278,096	60,381,831	2,103,735	MF56		
BW46	chr18	10,632,783	12,180,972	1,548,189	MF58		*impossible*

Table 5.6: mrFAST inversions compared against BWA inversions, InvFEST and the callset

MF ID	chrom	inner start	inner end	size	BW ID	InvFEST	callset result
MF1	chr1	144,828,037	145,370,642	542,605			
MF2	chr2	87,263,205	87,384,777	121,572	BW1	HsInv0242	
MF3	chr2	91,985,955	92,252,738	266,783	BW2		
MF4	chr2	130,824,289	131,206,878	382,589			
MF5	chr7	64,624,518	64,998,550	374,032	BW3	HsInv0484	
MF6	chr7	64,971,084	65,115,077	143,993		HsInv0484	
MF7	chr7	143,895,393	144,046,137	150,744			
MF8	chr9	46,696,377	46,834,953	138,576			
MF9	chr10	42,382,707	42,535,677	152,970	BW4		
MF10	chr16	14,877,924	15,441,797	563,873	BW7	HsInv0551	
MF11	chr16	21,456,067	21,883,232	427,165	BW8		
MF12	chr16	21,482,090	21,931,837	449,747			
MF13	chr16	69,978,097	70,208,270	230,173	BW9		

Continued on next page

Table 5.6 – *Continued from previous page (mrFAST vs. BWA)*

MF ID	chrom	inner start	inner end	size	BW ID	InvFEST	callset result
MF14	chr17	44,369,719	44,586,239	216,520	BW11		
MF15	chr17	44,400,349	44,609,394	209,045	BW11		
MF16	chr1	146,467,444	147,615,065	1,147,621			
MF17	chr2	96,499,871	97,833,880	1,334,009	BW12		
MF18	chr2	107,067,229	108,534,871	1,467,642	BW13		
MF19	chr2	111,280,588	113,186,168	1,905,580			*impossible*
MF20	chr2	130,826,487	132,112,334	1,285,847	BW14	HsInv0669 HsInv0698	
MF21	chr3	123,667,775	125,690,261	2,022,486	BW15		
MF22	chr5	21,508,067	29,438,065	7,929,998			
MF23	chr5	68,951,654	70,421,229	1,469,575	BW16		*not tested*
MF24	chr5	175,692,620	177,170,160	1,477,540	BW17	HsInv0687 HsInv0273 HsInv0281 HsInv0280 HsInv0276	
MF25	chr7	32,868,354	35,028,527	2,160,173	BW18		
MF26	chr7	51,457,375	56,451,750	4,994,375	BW19		
MF27	chr7	66,641,494	72,027,054	5,385,560	BW20		
MF28	chr7	72,593,223	74,358,790	1,765,567	BW21		
MF29	**chr8**	**7,020,318**	**12,480,229**	**5,459,911**	**BW22**	**HsInv0501** **HsInv0496** **HsInv0713** **HsInv0717** **HsInv0494** **HsInv0497**	
MF30	chr9	86,441,747	88,382,909	1,941,162	BW23		
MF31	chr9	97,062,107	99,672,076	2,609,969	BW24		
MF32	chr10	81,460,982	89,005,838	7,544,856	BW25		
MF33	chr14	19,427,037	20,134,688	707,651	BW27		
MF34	chr14	63,584,607	67,140,192	3,555,585			
MF35	chr15	23,391,111	28,625,879	5,234,768	BW29		*not confirmed*

Continued on next page

Table 5.6 – *Continued from previous page (mrFAST vs. BWA)*

MF ID	chrom	inner start	inner end	size	BW ID	InvFEST	callset result
MF36	**chr15**	**30,825,273**	**32,850,269**	**2,024,996**	**BW31**	**HsInv1049**	***confirmed***
MF37	chr15	74,360,818	75,595,294	1,234,476	BW32	HsInv0544 HsInv0771	
MF38	**chr15**	**83,009,461**	**84,835,661**	**1,826,200**	**BW33**	**HsInv0547**	***confirmed***
MF39	chr15	100,348,404	102,336,630	1,988,226	BW34		
MF40	chr16	12,021,483	21,893,378	9,871,895			
MF41	chr16	15,029,573	16,324,009	1,294,436	BW36		
MF42	chr16	15,249,321	18,418,056	3,168,735			
MF43	chr16	16,649,631	18,715,257	2,065,626	BW37		
MF44	chr16	18,933,343	21,512,037	2,578,694			
MF45	chr16	21,455,882	30,336,523	8,880,641			*not confirmed*
MF46	chr16	21,758,040	22,575,584	817,544	BW38		
MF47	chr16	21,841,285	29,532,007	7,690,722	BW39		
MF48	chr16	22,509,013	28,651,702	6,142,689			
MF49	chr16	31,911,754	33,750,338	1,838,584			*not tested*
MF50	chr16	32,328,258	33,354,611	1,026,353			
MF51	chr16	70,221,140	74,483,623	4,262,483	BW41		
MF52	chr17	15,474,088	18,569,007	3,094,919	BW42		*not confirmed*
MF53	chr17	18,479,530	20,310,049	1,830,519	BW43		
MF54	chr17	36,226,733	45,524,595	9,297,862			
MF55	chr17	43,605,052	45,153,801	1,548,749			
MF56	chr17	58,278,096	60,381,831	2,103,735	BW45		
MF57	chr18	5,329,111	13,978,445	8,649,334			
MF58	chr18	10,585,261	12,201,977	1,616,716	BW46		*impossible*

Table 5.7: InvFEST inversions on NA12878 that could be lifted over to hg19 compared against inversions called by dipSeq.

InvFEST	status	chrom	start–end	size	MF ID	BW ID
HsInv0431	U (1/1)	chr1	16,863,917-17,015,524	151,607		
HsInv0427	P (1/1)	chr1	16,967,642-17,090,053	122,411		
HsInv0233	U (2/2)	chr1	108,758,451-109,023,381	264,930		
HsInv0438	U (1/1)	chr1	145,291,523-148,026,038	2,734,515		
HsInv0662	U (1/1)	chr1	146,459,383-147,604,990	1,145,607		
HsInv0659	U (1/1)	chr1	248,596,855-248,822,042	225,187		
HsInv0242	U (1/2)	chr2	86,924,134-87,332,666	408,532	MF2	BW1
HsInv0251	U (1/1)	chr2	97,860,872-98,165,105	304,233		
HsInv0669	P (1/1)	chr2	130,882,737-132,306,424	1,423,687	MF20	BW14
HsInv0698	U (1/1)	chr2	130,937,621-132,258,916	1,321,295	MF20	BW14
HsInv0462	P (1/1)	chr3	195,316,136-197,435,840	2,119,704		
HsInv0465	U (1/2)	chr3	195,436,065-195,764,840	328,775		
HsInv0687	U (1/1)	chr5	175,324,953-177,362,931	2,037,978	MF24	BW17
HsInv0273	U (1/1)	chr5	175,388,492-177,317,608	1,929,116	MF24	BW17
HsInv0281	U (1/1)	chr5	175,417,374-177,297,393	1,880,019	MF24	BW17
HsInv0280	U (1/1)	chr5	175,420,920-177,265,063	1,844,143	MF24	BW17
HsInv0276	U (1/1)	chr5	175,661,037-177,097,205	1,436,168	MF24	BW17
HsInv0277	U (1/1)	chr5	177,170,537-177,462,805	292,268		
HsInv0290	V (×)	chr7	5,933,270-6,872,518	939,248		
HsInv0287	U (1/1)	chr7	57,676,801-57,899,569	222,768		
HsInv0484	U (1/1)	chr7	65,019,296-65,118,708	99,412	MF6	BW3
HsInv0708	U (1/1)	chr7	77,485,837-86,158,162	8,672,325		
HsInv0702	P (1/2)	chr7	143,221,664-143,475,612	253,948	MF7	
HsInv0303	U (1/1)	chr7	143,221,829-143,509,828	287,999	MF7	
HsInv0489	U (1/1)	chr7	143,308,840-143,539,432	230,592	MF7	
HsInv0493	U (1/1)	chr7	149,560,947-152,534,515	2,973,568		
HsInv0710	U (1/1)	chr8	2,190,126-2,329,820	139,694		
HsInv0494	**P (1/1)**	**chr8**	**6,922,489-12,573,597**	**5,651,108**	**MF29**	**BW22**
HsInv0496	**P (1/1)**	**chr8**	**6,922,489-12,573,597**	**5,651,108**	**MF29**	**BW22**
HsInv0497	**P (1/1)**	**chr8**	**6,922,489-12,573,597**	**5,651,108**	**MF29**	**BW22**

Continued on next page

Table 5.7 – *Continued from previous page (InvFest vs dipSeq)*

InvFEST	status	chrom	start–end	size	MF ID	BW ID
HsInv0501	**P (1/1)**	**chr8**	**6,922,489-12,573,597**	**5,651,108**	**MF29**	**BW22**
HsInv0713	**P (1/1)**	**chr8**	**6,922,489-12,573,597**	**5,651,108**	**MF29**	**BW22**
HsInv0717	**P (1/1)**	**chr8**	**6,922,489-12,573,597**	**5,651,108**	**MF29**	**BW22**
HsInv0714	U (1/1)	chr8	7,597,184-7,876,379	279,195		
HsInv0498	U (1/1)	chr8	7,625,731-7,927,347	301,616		
HsInv0503	U (1/2)	chr9	90,534,956-90,753,046	218,090		
HsInv0739	U (1/1)	chr10	51,577,574-51,738,575	161,001		
HsInv0330	U (2/3)	chr11	50,078,702-50,411,307	332,605		BW5
HsInv0333	U (1/1)	chr11	89,530,160-89,789,619	259,459		
HsInv0743	P (1/1)	chr11	89,576,361-89,727,104	150,743		
HsInv0751	U (1/3)	chr12	9,403,662-9,624,067	220,405		
HsInv0149	P (1/1)	chr13	25,571,995-26,650,325	1,078,330		
HsInv0759	P (1/4)	chr13	52,905,750-53,077,057	171,307		BW67
HsInv1049	**V (T)**	**chr15**	**30,370,112-32,899,708**	**2,529,596**	**MF56**	**BW31**
HsInv0151	P (1/1)	chr15	45,111,375-45,377,413	266,038		
HsInv0545	U (1/1)	chr15	45,130,002-45,363,863	233,861		
HsInv0771	U (1/1)	chr15	74,344,087-75,615,381	1,271,294	MF37	BW32
HsInv0544	V (F)	chr15	74,352,986-75,597,139	1,244,153	MF37	BW32
HsInv0547	**U (1/1)**	**chr15**	**82,996,464-84,931,502**	**1,935,038**	**MF38**	**BW33**
HsInv0789	U (1/1)	chr16	2,561,305-2,730,330	169,025		
HsInv0551	U (1/1)	chr16	15,009,740-15,128,775	119,035	MF10	
HsInv0152	U (1/2)	chr16	21,593,897-22,712,112	1,118,215	MF46	BW38
HsInv0557	U (1/1)	chr16	32,067,714-33,666,016	1,598,302	MF50	
HsInv0555	U (1/1)	chr16	32,289,960-32,667,352	377,392	MF50	
HsInv0364	U (1/1)	chr16	32,880,787-33,794,210	913,423	MF50	
HsInv0553	U (1/1)	chr16	34,390,796-34,764,406	373,610		
HsInv0791	U (1/3)	chr17	2,953,435-3,153,225	199,790		
HsInv0373	U (1/1)	chr17	16,731,050-18,331,974	1,600,924		
HsInv1051	V (F)	chr17	18,501,299-18,751,409	250,110		
HsInv1048	**V (F*)**	**chr17**	**34,725,850-36,295,000**	**1,569,150**		**BW44**
HsInv0573	V	chr17	43,573,203 -44,784,489	1,211,286	MF55	
HsInv0382	U (1/2)	chr20	25,724,734-26,094,266	369,532		

Continued on next page

Table 5.7 – *Continued from previous page (InvFest vs dipSeq)*

InvFEST	status	chrom	start–end	size	MF ID	BW ID
HsInv0385	U (1/1)	chr22	18,722,345-18,868,134	145,789		
HsInv0809	U (1/1)	chr22	20,609,431-21,579,386	969,955		
HsInv0592	U (1/1)	chr22	21,492,639-21,632,913	140,274		
HsInv0595	U (1/1)	chr22	21,711,123-21,941,996	230,873		
HsInv0807	U (1/1)	chr22	21,741,762-21,895,904	154,142		
HsInv0605	P (1/1)	chrX	51,800,553-51,939,004	138,451		
HsInv0831	P (1/1)	chrX	101,443,608-101,744,661	301,053		
HsInv0610	U (2/3)	chrX	134,222,760-134,426,178	203,418		
HsInv0815	U (1/1)	chrX	152,350,907-152,572,142	221,235		
HsInv0598	U (1/1)	chrX	152,400,875-152,541,422	140,547		
HsInv0408	U (1/1)	chrX	152,401,911-152,523,821	121,910		

The InvFEST status is the status given in the InvFEST.

Status is U: unreliable prediction, V: validated (result in paranthesis), P: predicted .

The numbers given in the parentheses (x/y) are:

x: number of studies that predict the inversion

y: total number of studies.

* This inversion has been validated in another study [68].

($^{\times}$) The inversion has been predicted by one study and not predicted in the other but the validation result is not provided for NA12878.

5.4 The validated call set

We selected a total of 12 inversions with high support for experimental validation from both callsets, to represent shared, BWA-specific, and mrFAST-specific calls (Table 5.8) and compared our predictions with the known inversions reported in the InvFEST database [41], and found that dipSeq could correctly identify all three inversions that are previously *validated* in the genome of the same individual; a 5 Mbp inversion in 8p23.1 [68], a 1.5 Mbp inversion in 17q12 [68], and a 2 Mbp inversion in 15q13.3 [69]. Out of the remaining 8 inversion predictions, 2 could not be tested due to the segmental duplications around the breakpoints. We tested the remaining using FISH experiments, and validated a novel inversion in the 15q25 locus (Figure 5.4a,b) which breaks the GOLGA8

gene on one breakpoint. We also show the visualization of a previously characterized 15q13 inversion (InvFEST ID: HsInv1049) using the SAVANT browser [70] in Figure 5.4c. The tested breakpoints and the result and which method they were provided by are given in Table]reftab:callset. The segmental duplications and gaps around the breakpoints of each inversion are depicted in figures 5.5 to 5.13[2].

Table 5.8: Summary of validation of inversions predicted in the genome of NA12878 using dipSeq.

ID	chrom	start–end	result	MF ID	BW ID	InvFEST ID
CS1	chr2	110,887,269–113,351,503	incomplete	MF19	–	
CS2	chr5	69,080,890–70,004,538	not performed	MF23	BW16	
CS3	chr8	6,922,489–12,573,597	confirmed [68]$^{\times}$	MF56	BW31	HsInv0501
CS4	chr14	19,369,507–20,154,427	not performed	MF33	BW26 BW27	
CS5	chr15	22,667,129–28,772,134	not confirmed	MF35	BW29	
CS6	chr15	30,370,112–32,899,708	confirmed [69]$^{\times}$	MF56	BW31	HsInv1049
CS7	chr15	83,290,936–84,688,129	confirmed*	MF38	BW33	HsInv0547
CS8	chr16	21,847,556–30,283,910	not confirmed	MF47	BW39	
CS9	chr16	32,277,947–33,295,746	not performed	MF50	–	HsInv0364
CS10	chr17	15,544,928–18,621,866	not confirmed	MF52	BW42	
CS11	chr17	34,725,850–36,295,000	confirmed [68]	–	BW44	HsInv1048
CS12	chr18	10,668,776–12,210,270	incomplete	MF58	BW46	

incomplete : The test was not completed because the probes mapped to different chromosome due to repeats.
not performed: The inversion was not tested because the probes were not available in the prepared clone library [28].
MF ID and BW ID are the unique IDs of the inversions predicted by dipSeq given in Tables 5.1 to 5.4.
* Validated by FISH experiments using the preprepared fosmid clone libraries of [28].
$^{\times}$ Validated in the InvFEST database.

[2]Obtained from the UCSC genome browser

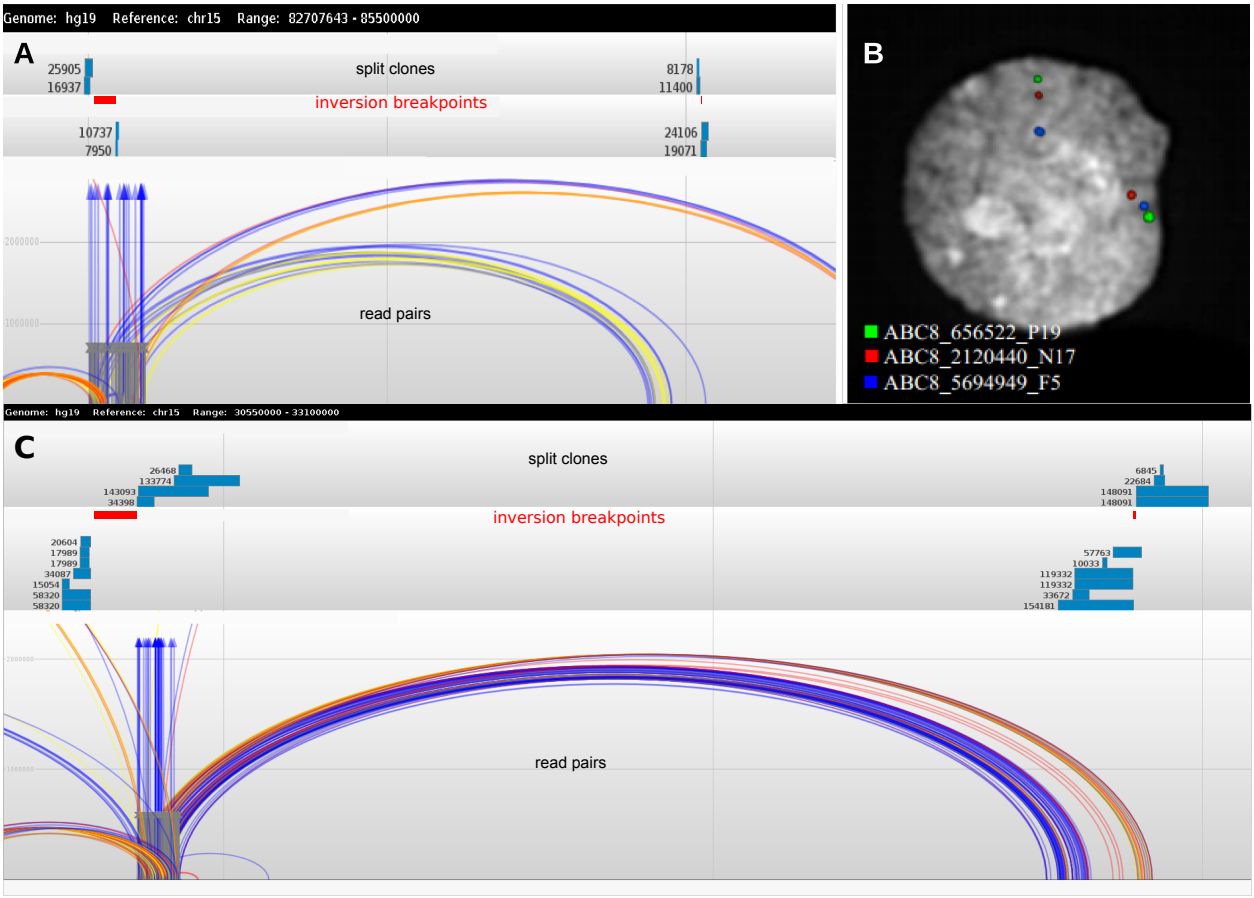


Figure 5.4: Inversions discovered by dipSeq in the NA12878 genome.

(A) Novel inversion found at chr15:83,089,659-84,865,500 (inner coordinates). We show the locations of split clones and the supporting read pairs using the SAVANT browser [70]. (B) Experimental validation of the novel inversion discovered using interphase FISH (green-red-blue: direct, green-blue-red:inverted). (C) SAVANT browser view of the previously known inversion at chr15:30,433,406-32,898,559. SAVANT read pair colors are as follows. Light blue: concordant, red: discordant by length, dark blue: one end inverted, yellow: everted (tandem duplication), gray: one end unmapped.

5.4.1 Visualization

In the Figures 5.5, Figure 5.6, Figure 5.7, Figure 5.8, Figure 5.9, Figure 5.10, Figure 5.11, Figure 5.12, Figure 5.13, Figure 5.14, Figure 5.15, and Figure 5.16 show the location of the 12 inversions given in Table tab:callset. Breakpoints predicted by dipSeq are displayed with black regions at the very top along with the assembly gaps and segmental duplications around them. Images were taken from the UCSC genome browser with 1.5X zoom out.

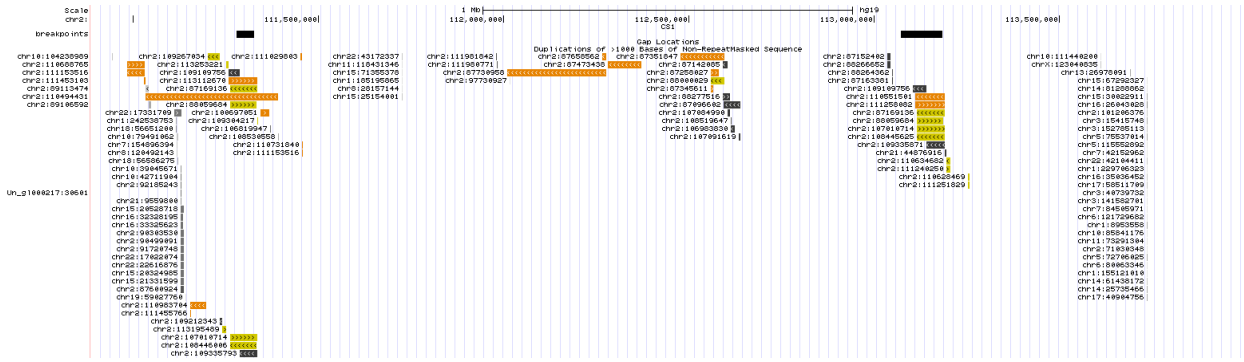


Figure 5.5: Segmental duplications around the breakpoints of CS1 given in Table 5.8

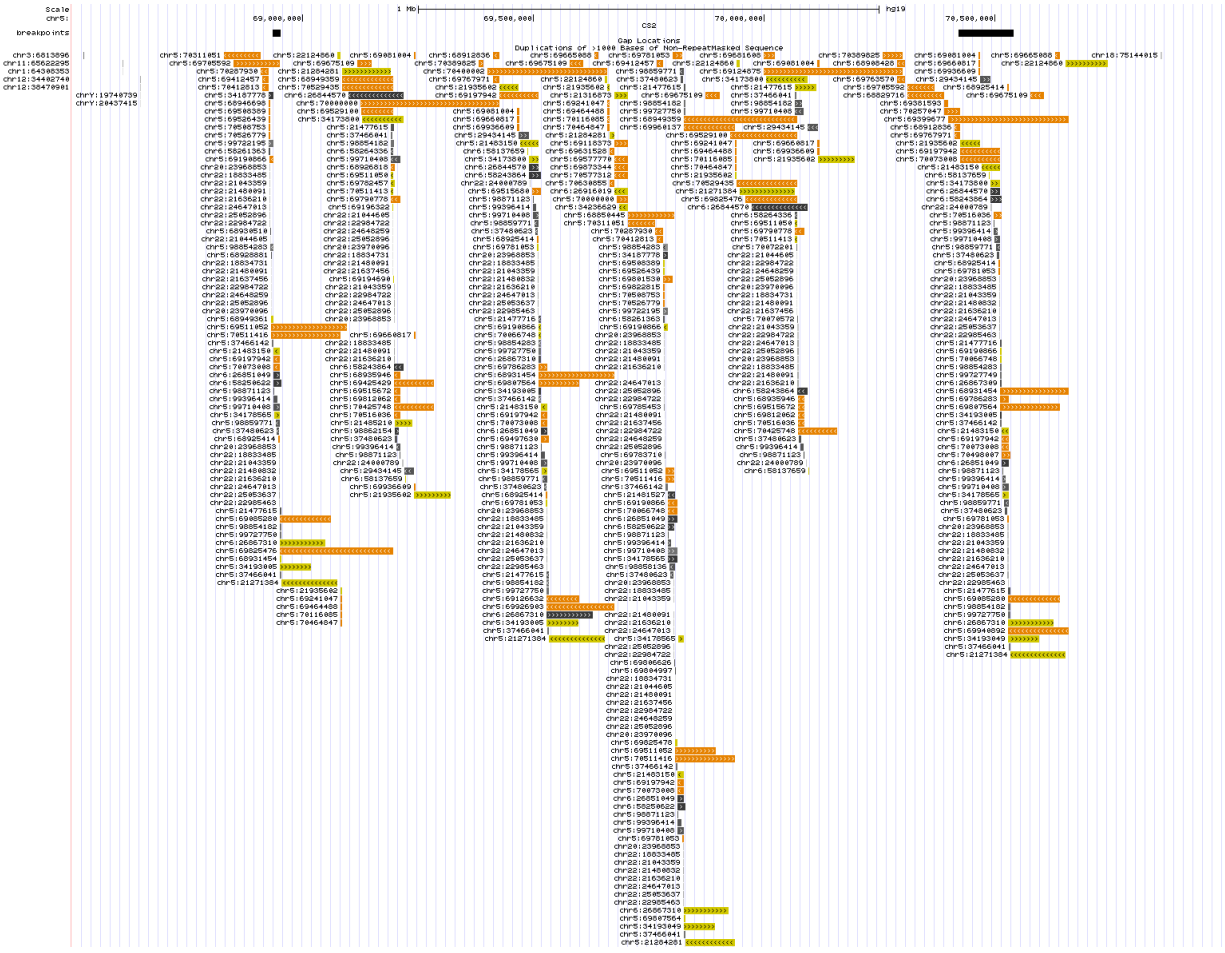
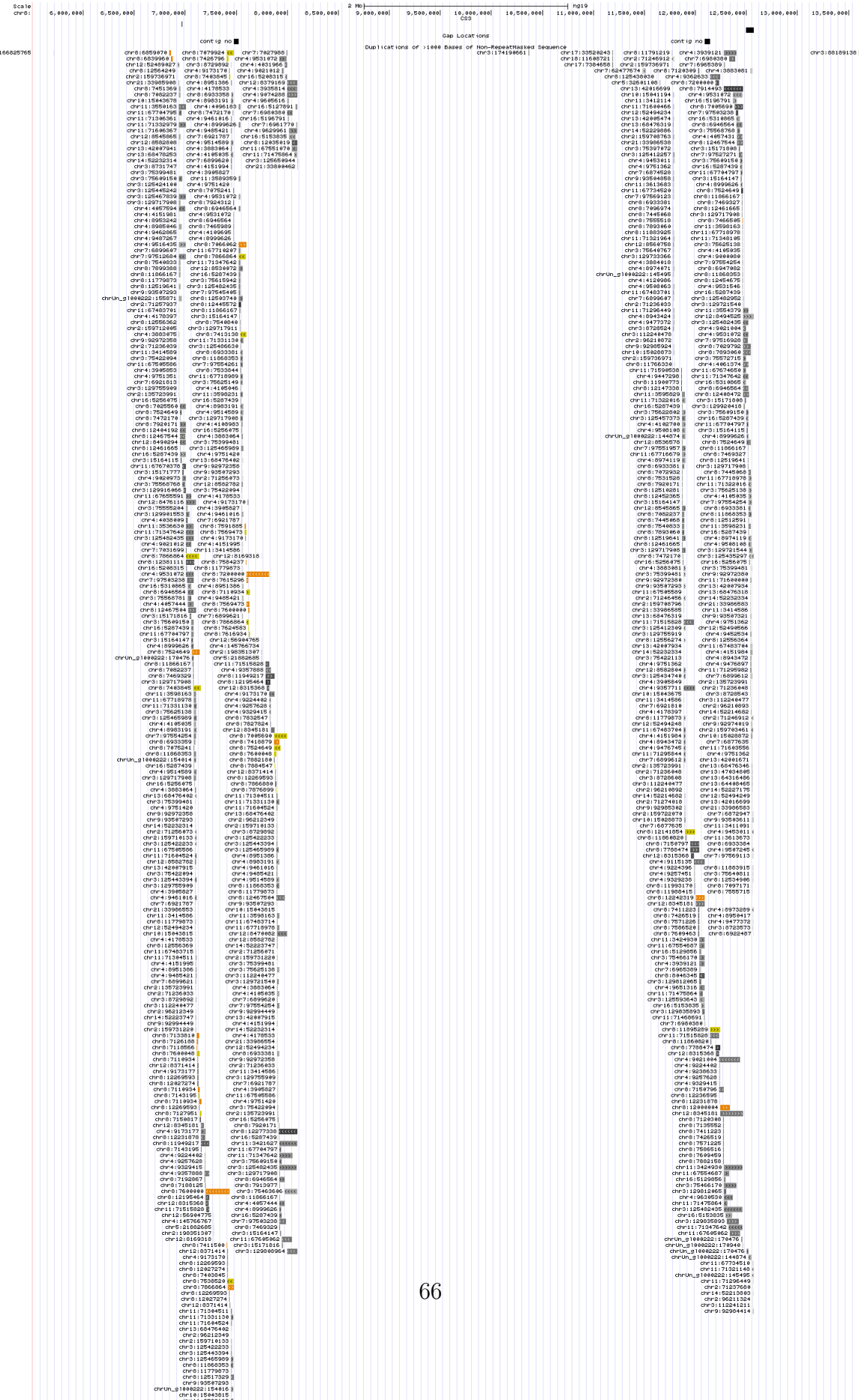


Figure 5.6: Segmental duplications around the breakpoints of CS2 given in Table 5.8


Scale
chr9:
6,000,000
6,500,000
7,000,000
7,500,000
8,000,000
8,500,000
9,000,000
9,500,000
10,000,000
10,500,000
11,000,000
11,500,000
12,000,000
12,500,000
13,000,000
13,500,000
2 Mb
hg19
CD3
Gap Locations
Duplications of >1000 Bases of Non-RepeatMasked Sequence
contig no
chr1:166625765
chr3:88169136

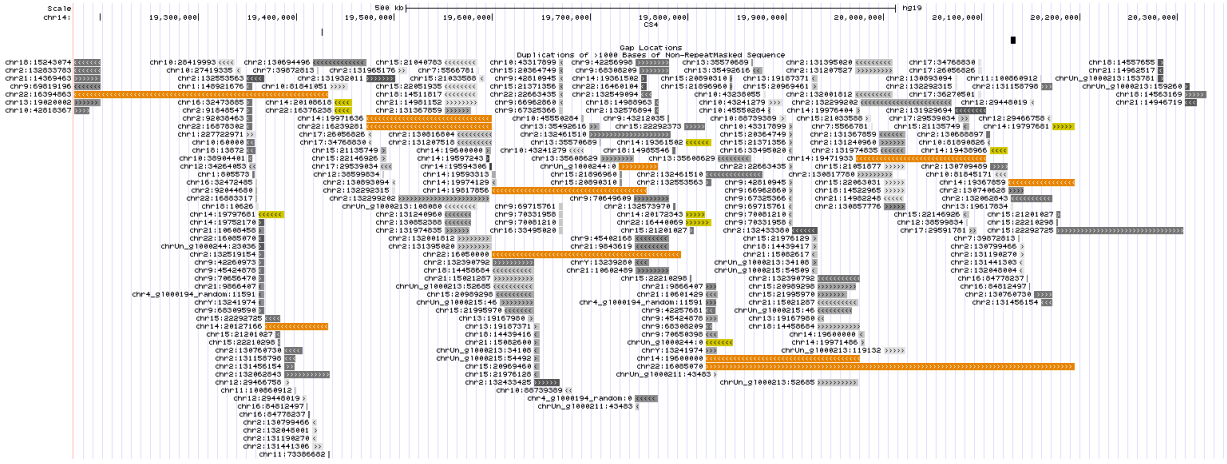


Figure 5.8: Segmental duplications around the breakpoints of CS4 given in Table 5.8

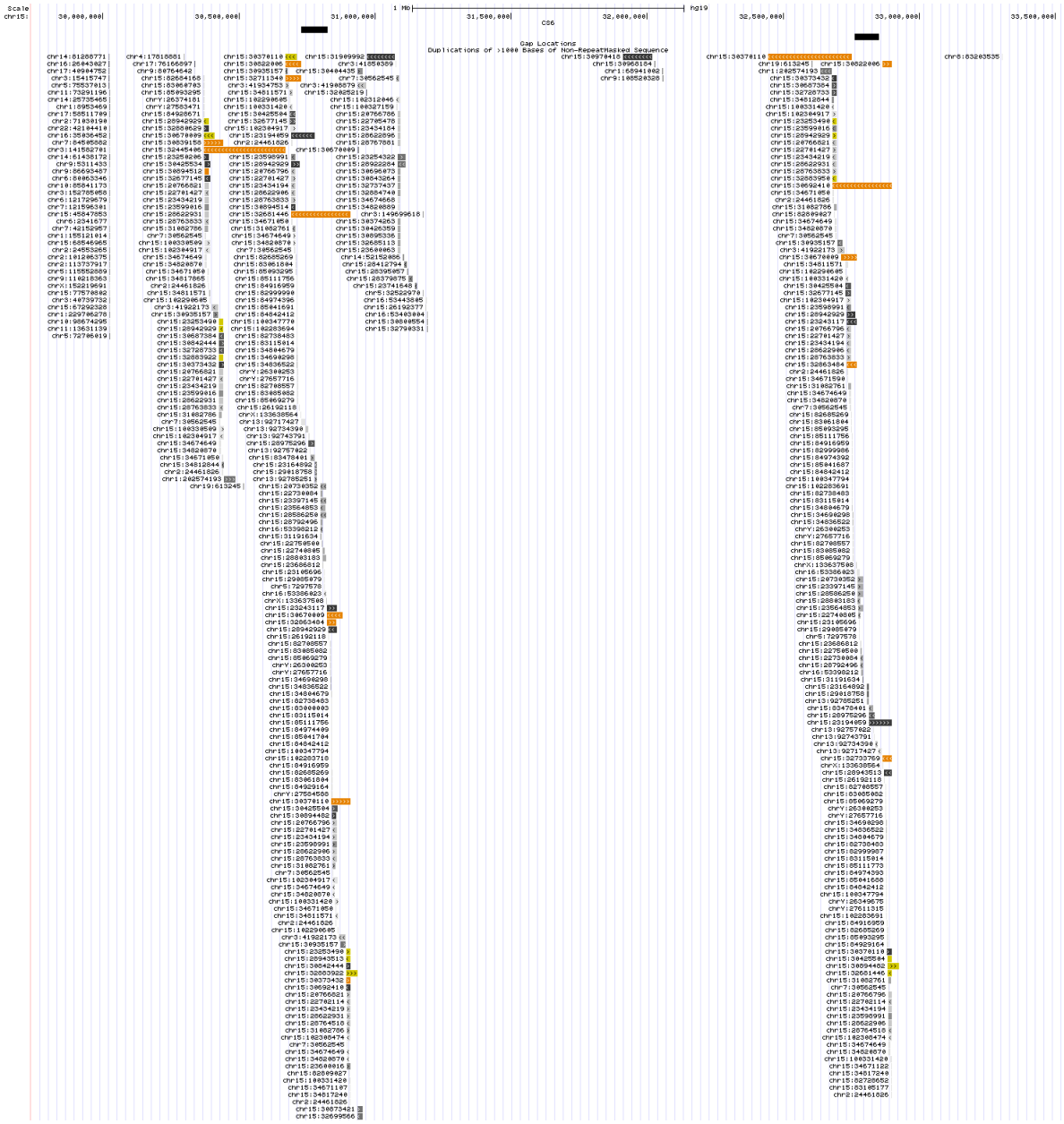


Figure 5.10: Segmental duplications around the breakpoints of CS6 given in Table 5.8

Figure 5.11: Segmental duplications around the breakpoints of CS7 given in Table 5.8

Figure 5.12: Segmental duplications around the breakpoints of CS8 given in Table 5.8

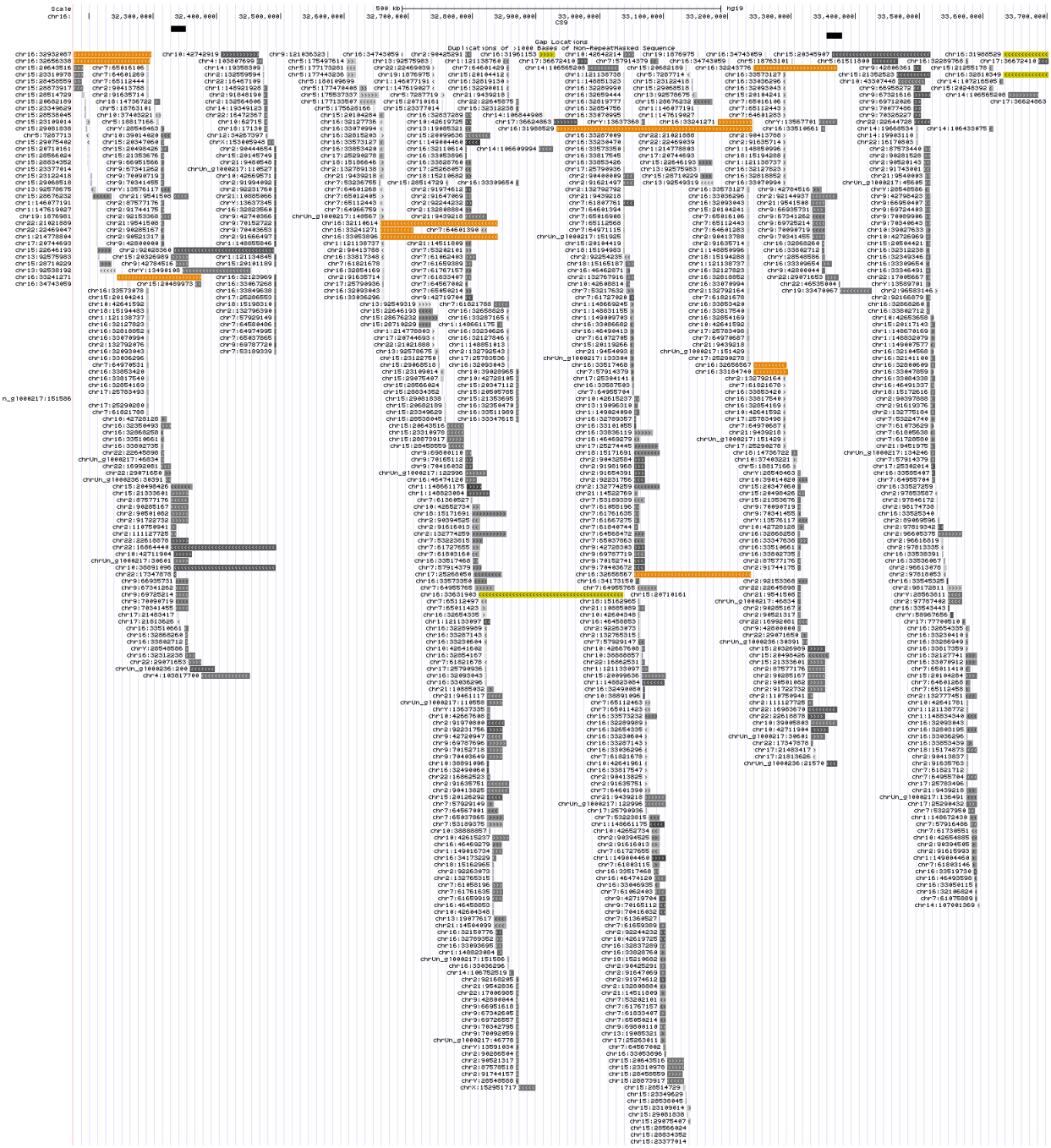


Figure 5.13: Segmental duplications around the breakpoints of CS9 given in Table 5.8

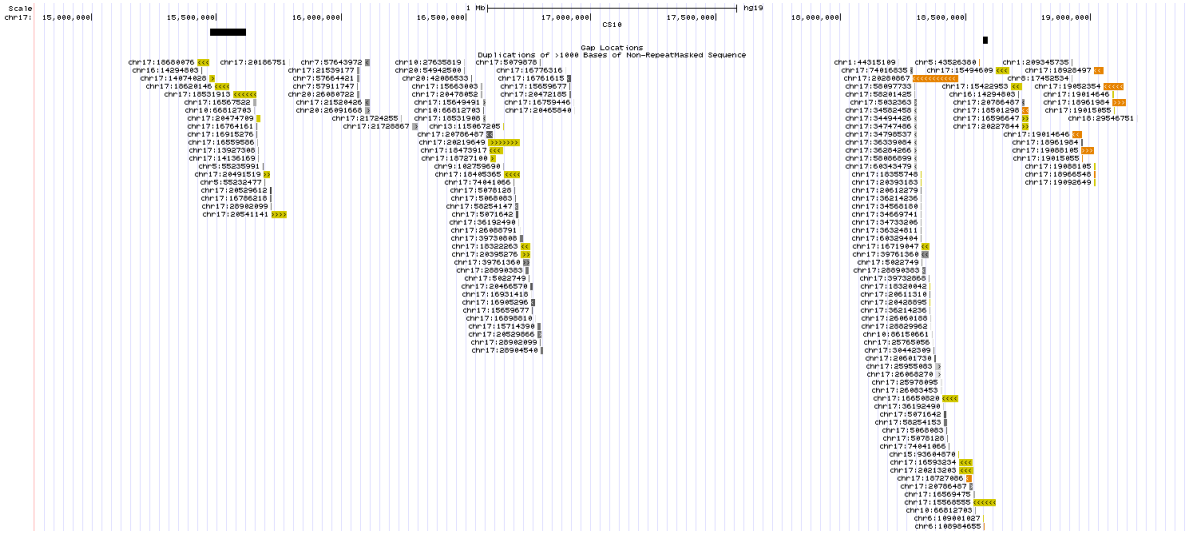


Figure 5.14: Segmental duplications around the breakpoints of CS10 given in Table 5.8

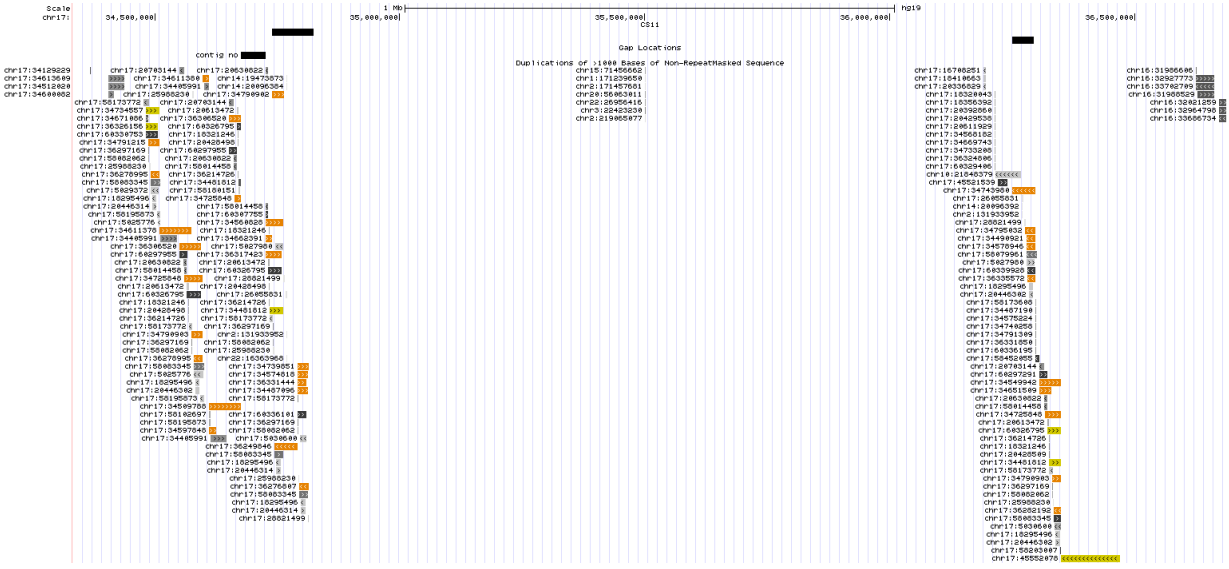


Figure 5.15: Segmental duplications around the breakpoints of CS11 given in Table 5.8

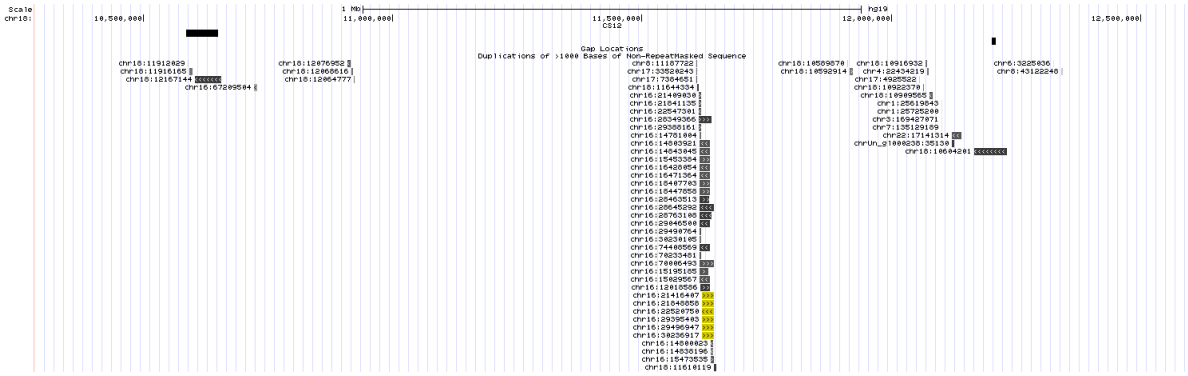


Figure 5.16: Segmental duplications around the breakpoints of CS12 given in Table 5.8

Chapter 6

Discussion

In summary, dipSeq is the first algorithm that can discover large genomic inversions using high throughput sequencing technologies. Our understanding of the phenotypic effects of inversions is still limited, and one of the reasons of this is the lack of reliable and cost effective methods to characterize such events. This is also true for other complex rearrangements such as duplications and translocations. Improvements in characterization of large complex rearrangements will help us better understand the biological mechanisms that lead to phenotypic difference, disease, and evolution.

In this thesis, we presented a novel algorithm, dipSeq, to characterize large genomic inversions using a new sequencing method initially developed to improve haplotype phasing. Although it suffers from high false positive rate using real data (Table 5.8), dipSeq was able to identify all previously validated inversion events, and also discover a novel variant. Furthermore, dipSeq performed better with simulated data, suggesting that the relatively poor performance with the NA12878 genome may be improved with higher depth of coverage.

6.1 Compatibility

dipSeq is *theoretically* compatible with all similarly constructed pooled sequence data, such as the TruSeq Synthetic Long-Reads (Moleculo) [23], or the Complete Genomics LFR Technology [24], provided that the pooled large DNA fragment sizes follow a Gaussian distribution. However, it should be noted that, large clone size is required to span segmental duplication blocks, and smaller clones such as fosmids may not be sufficient to detect inversions around segmental duplications [20]. Therefore, the theoretical minimum inversion size detectable by dipSeq is limited by clone length, i.e. 150 Kbp when BACs are used.

dipSeq is compatible with any aligner that can map paired end reads. The aligner can produce unique mappings (such as BWA) or all possible mappings (such as mrFAST). We have tested dipSeq on both aligners and no significant difference was observed.

From the computational perspective, dipSeq was tested on a system about 4GB RAM capacity, but obviously with larger data, more RAM is required. dipSeq runs on chromosomes one by one and does not keep much data in the memory. The main bottleneck is storing the read pairs which in the case of our real data required ∼2GB memory. In the case of the NA12878 data, dipsSeq ran in ∼17 minutes where most of the time required is for bedtools and bamtools to separate the reads. dipSeq is implemented in Java and used bamtools [71] and bedtools [67]. These programs should be installed on the system prior to running dipSeq.

6.2 Restrictions

dipSeq's ability to detect inversions is restricted by the data it is provided. dipSeq relies on the several statistical facts:

1. Clone sizes should come from a *Gaussian* distribution with a mean and standard deviation and a lower cutoff all three parameters should be given to dipSeq.
2. The clones should be distributed *uniformly* into a constant number of pools

3. The clones should cover the genome *uniformly* with a constant physical coverage rate.

4. The clones should be sequenced *uniformly* with a constant error and sequencing coverage.

5. The clones in each pool do not overlap (or the overlap rate is low).

6.2.1 The detectable size

The minimum detectable size which is detectable by dipSeq is limited to the average clone size. However in order to have reliable predictions we should allow for few clone sizes. There is no limitation on the upper bound. But if the size is too large, after finding the breakpoints, it is advised to check for gaps around the breakpoints. Also stratifying the search range can give us a better understanding of the SVs inside the breakpoints and help us avoid them.

6.2.2 Discovery of an inversion

dipSeq ability to discover an inversion with breakpoints B1 and B2 depends on first, in one pool there exists only one clone compassing B1 and does not exist any other clone compassing B2; and second, in another pool there exists only one clone compassing B2 and does not exist any other clone compassing B1. In this case we call the inversion discoverable. Yet this condition is necessary but not sufficient to find the inversion. The third condition that must hold is that these two clones must be reconstructable, meaning there should be enough sequences from those clones (at least 50% covered).

6.2.3 Low physical coverage

Lowering the physical coverage (number of clones per pool) reduces the true positive discovery rate. Obviously when there is not enough physical coverage (not enough clones) the

probability of having clones spanning the breakpoints decreases. We tried to calculate the probability of discovering an inversion with dipSeq given a constant physical coverage of a given genome length in a number of pools. This calculation is NP-complete ignoring the complication of having several inversions and the percentage of repeated regions (See Appendix A.3). However even if there exists clones that intersect the inversion breakpoint, if the sequencing coverage (average depth of reads per clone) is low, the clone might not be discoverable. In our study we noticed that due to the heuristic approach of clone reconstruction, dipSeq does not require very high sequencing coverage. But reducing the physical coverage defects the inversion discovery.

6.2.4 High physical coverage

On the other hand, high physical coverage will increase the probability of having overlapping clones in one pool. If in one pool two overlapping clones compass an inversion breakpoint, that breakpoint will not be detected by dipSeq (unless it is detectable in other pools) because the split clones will not have an abnormal size. However by increasing the standard deviation of clone size, keeping the sequencing coverage relatively low, dipSeq can still find these larger split clones in exchange for longer execution time and more false positive discovery rate. Increasing the standard deviation increases the false *split clone* discovery rate which should be normally discarded due to no read support. Here if the sequencing coverage is too high, due to the relative increase in sequencing errors (A instead of T) and mapping error (SDs and repeats), more false read support will be produced and more false split clones will be supported instead of discarded. This problem is negligible in simulation data where the sequencing errors are uniformly distributed, but in real data most false positives are due to such erroneous reads. In our experiments we observed 3-5X physical coverage is enough for dipSeq to detect inversions. We did not test higher coverage because it's not affordable to produce pooled clone data is such high coverage.

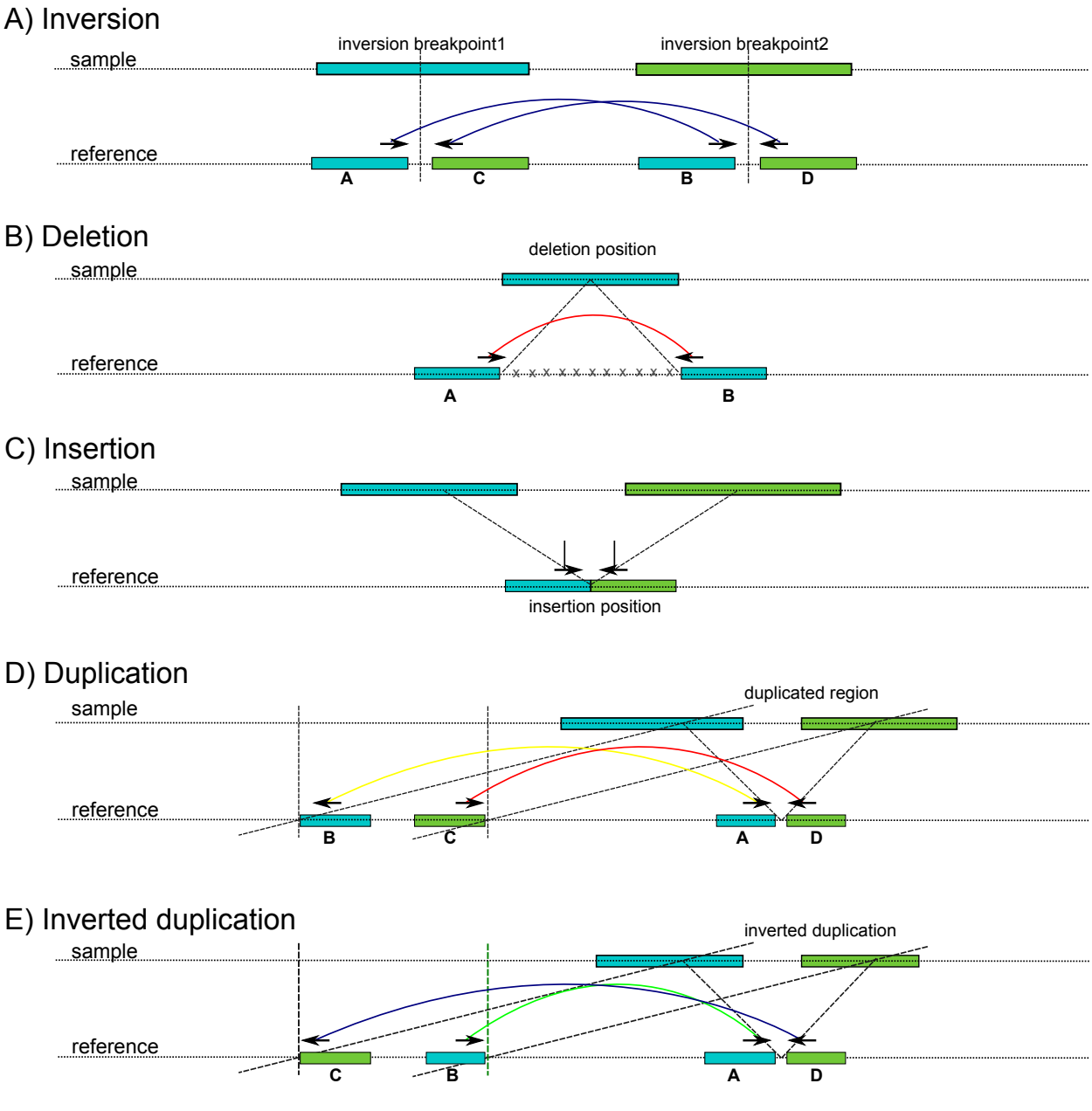


Figure 6.1: Split clone signal for other types of SV.

6.3 Future work

There are multiple directions that we can take to further improve dipSeq. First, to reduce the false discovery rate, we can incorporate split read sequence signature [21], and we can perform local *de novo* assembly around the predicted breakpoint intervals with an approach similar to TIGRA [60]. However, since both of these methods need high sequence coverage, they might not be suitable to directly apply to the low-coverage data set we used. Instead, it will be better to simultaneously use WGS data generated from the genome of the same individual. Since the PCS method also requires WGS data for haplotype phasing, it can be expected to generate matching PCS-WGS data sets from the same genomes.

Another future research on dipSeq will be testing and improving its abilities to discover smaller, yet still large inversions (>100 Kbp). In the course of this thesis, we focused on inversions larger than 500 Kbp, because the upper size limit for GASVPro [18] algorithm is 500 Kbp, and only such large inversions can be reliably tested using FISH. Note that validating smaller inversions is a more difficult task, using fiber FISH, or PCR if the breakpoints lie within unique regions. In addition, the clone size distribution should be tighter to ensure clone reconstruction method does not artificially "merge" split clones into a single interval. Alternatively, we can try to use smaller clones such as fosmids, despite their limitations. We still would like to investigate dipSeq's performance using real fosmid data, however, this may require additional algorithmic enhancements especially in the presence of nearby segmental duplications [20]. There is currently only one pooled fosmid sequencing dataset [20] generated from the genome of a Gujarati Indian individual (NA20847). We would like to apply dipSeq to the NA20847 dataset and evaluate its performance with experimental validation.

dipSeq can also be extended to characterize other forms of large structural variation, including deletions, insertions, direct and inverted duplications. Each of these types of SV present themselves with different split clone signatures that we summarize in Figure 6.1. We also note that, determining the location of a segmental duplication event is yet a largely unsolved problem, even when long reads are used [17]. It may also be possible to discover translocations using split clones, however, chance of finding incorrect split

clones will also increase, causing a reduction in the performance of maximal quasi clique approximation.

6.4 Funding

Funding for this project was provided by a Marie Curie Career Integration Grant (303772) and an EMBO grant (IG-2521) to C.A., an NIH grant (HG004120) to E.E.E., and a Firb-Programma "Futuro in Ricerca" grant (RBFR103CE3) to M.V. C.A. also acknowledges support from The Science Academy of Turkey, under the BAGEP program. **Conflict of Interest:** E.E.E. is on the scientific advisory board for DNAnexus, Inc.

Bibliography

[1] C. Alkan, B. P. Coe, and E. E. Eichler, "Genome structural variation discovery and genotyping.," *Nat Rev Genet*, vol. 12, pp. 363–376, May 2011.

[2] F. E. Dewey, S. Pan, M. T. Wheeler, S. R. Quake, and E. A. Ashley, "Dna sequencing clinical applications of new dna sequencing technologies," *Circulation*, vol. 125, no. 7, pp. 931–944, 2012.

[3] A. J. Griffiths, W. M. Gelbart, J. H. Miller, and R. C. Lewontin, "Modern genetic analysis," 1999.

[4] M. Puig, S. Casillas, S. Villatoro, and M. Cáceres, "Human inversions and their functional consequences," *Briefings in functional genomics*, p. elv020, 2015.

[5] L. R. Osborne, M. Li, B. Pober, D. Chitayat, J. Bodurtha, A. Mandel, T. Costa, T. Grebe, S. Cox, L. C. Tsui, and S. W. Scherer, "A 1.5 million-base pair inversion polymorphism in families with Williams-Beuren syndrome.," *Nat Genet*, vol. 29, pp. 321–325, Nov 2001.

[6] G. Gimelli, M. A. Pujana, M. G. Patricelli, S. Russo, D. Giardino, L. Larizza, J. Cheung, L. Armengol, A. Schinzel, X. Estivill, and O. Zuffardi, "Genomic inversions of human chromosome 15q11-q13 in mothers of Angelman syndrome patients with class ii (bp2/3) deletions.," *Hum Mol Genet*, vol. 12, pp. 849–858, Apr 2003.

[7] R. Visser, O. Shimokawa, N. Harada, N. Niikawa, and N. Matsumoto, "Non-hotspot-related breakpoints of common deletions in sotos syndrome are located within destabilised dna regions.," *J Med Genet*, vol. 42, p. e66, Nov 2005.

[8] H. Stefansson, A. Helgason, G. Thorleifsson, V. Steinthorsdottir, G. Masson, J. Barnard, A. Baker, A. Jonasdottir, A. Ingason, V. G. Gudnadottir, N. Desnica, A. Hicks, A. Gylfason, D. F. Gudbjartsson, G. M. Jonsdottir, J. Sainz, K. Agnarsson, B. Birgisdottir, S. Ghosh, A. Olafsdottir, J.-B. Cazier, K. Kristjansson, M. L. Frigge, T. E. Thorgeirsson, J. R. Gulcher, A. Kong, and K. Stefansson, "A common inversion under selection in Europeans.," *Nat Genet*, vol. 37, pp. 129–137, Feb 2005.

[9] A. J. Sharp, Z. Cheng, and E. E. Eichler, "Structural variation of the human genome," *Annu Rev Genomics Hum Genet*, vol. 7, pp. 407–442, 2006.

[10] D. A. Koolen, L. E. L. M. Vissers, R. Pfundt, N. de Leeuw, S. J. L. Knight, R. Regan, R. F. Kooy, E. Reyniers, C. Romano, M. Fichera, A. Schinzel, A. Baumer, B.-M. Anderlid, J. Schoumans, N. V. Knoers, A. G. van Kessel, E. A. Sistermans, J. A. Veltman, H. G. Brunner, and B. B. A. de Vries, "A new chromosome 17q21.31 microdeletion syndrome associated with a common inversion polymorphism.," *Nat Genet*, vol. 38, pp. 999–1001, Sep 2006.

[11] M. C. Zody, Z. Jiang, H.-C. Fung, F. Antonacci, L. W. Hillier, M. F. Cardone, T. A. Graves, J. M. Kidd, Z. Cheng, A. Abouelleil, L. Chen, J. Wallis, J. Glasscock, R. K. Wilson, A. D. Reily, J. Duckworth, M. Ventura, J. Hardy, W. C. Warren, and E. E. Eichler, "Evolutionary toggling of the *mapt* 17q21.31 inversion region.," *Nat Genet*, vol. 40, pp. 1076–1083, Sep 2008.

[12] K. M. Steinberg, F. Antonacci, P. H. Sudmant, J. M. Kidd, C. D. Campbell, L. Vives, M. Malig, L. Scheinfeldt, W. Beggs, M. Ibrahim, G. Lema, T. B. Nyambo, S. A. Omar, J.-M. Bodo, A. Froment, M. P. Donnelly, K. K. Kidd, S. A. Tishkoff, and E. E. Eichler, "Structural diversity and african origin of the 17q21.31 inversion polymorphism.," *Nat Genet*, vol. 44, pp. 872–880, Aug 2012.

[13] P. Medvedev, M. Stanciu, and M. Brudno, "Computational methods for discovering structural variation with next-generation sequencing.," *Nat Methods*, vol. 6, pp. S13–S20, Nov 2009.

[14] F. Hormozdiari, C. Alkan, E. E. Eichler, and S. C. Sahinalp, "Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes.," *Genome Res*, vol. 19, pp. 1270–1278, Jul 2009.

[15] T. Rausch, T. Zichner, A. Schlattl, A. M. Stütz, V. Benes, and J. O. Korbel, "Delly: structural variant discovery by integrated paired-end and split-read analysis," *Bioinformatics*, vol. 28, no. 18, pp. i333–i339, 2012.

[16] R. M. Layer, C. Chiang, A. R. Quinlan, and I. M. Hall, "LUMPY: a probabilistic framework for structural variant discovery.," *Genome Biology*, vol. 15, p. R84, Jun 2014.

[17] M. J. P. Chaisson, J. Huddleston, M. Y. Dennis, P. H. Sudmant, M. Malig, F. Hormozdiari, F. Antonacci, U. Surti, R. Sandstrom, M. Boitano, J. M. Landolin, J. A. Stamatoyannopoulos, M. W. Hunkapiller, J. Korlach, and E. E. Eichler, "Resolving the complexity of the human genome using single-molecule sequencing.," *Nature*, vol. 517, pp. 608–611, Jan 2015.

[18] S. S. Sindi, S. Onal, L. C. Peng, H.-T. Wu, and B. J. Raphael, "An integrative probabilistic model for identification of structural variation in sequencing data.," *Genome Biol*, vol. 13, no. 3, p. R22, 2012.

[19] International Human Genome Sequencing Consortium, "Initial sequencing and analysis of the human genome.," *Nature*, vol. 409, pp. 860–921, Feb 2001.

[20] J. O. Kitzman, A. P. Mackenzie, A. Adey, J. B. Hiatt, R. P. Patwardhan, P. H. Sudmant, S. B. Ng, C. Alkan, R. Qiu, E. E. Eichler, and J. Shendure, "Haplotype-resolved genome sequencing of a Gujarati Indian individual.," *Nat Biotechnol*, vol. 29, pp. 59–63, Jan 2011.

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

[22] M. Brunato, H. H. Hoos, and R. Battiti, "On effectively finding maximal quasi-cliques in graphs," in *Learning and Intelligent Optimization*, pp. 41–55, Springer, 2008.

[23] V. Kuleshov, D. Xie, R. Chen, D. Pushkarev, Z. Ma, T. Blauwkamp, M. Kertesz, and M. Snyder, "Whole-genome haplotyping using long reads and statistical methods.," *Nat Biotechnol*, vol. 32, pp. 261–266, Mar 2014.

[24] B. A. Peters, B. G. Kermani, A. B. Sparks, O. Alferov, P. Hong, A. Alexeev, Y. Jiang, F. Dahl, Y. T. Tang, J. Haas, K. Robasky, A. W. Zaranek, J.-H. Lee, M. P. Ball, J. E. Peterson, H. Perazich, G. Yeung, J. Liu, L. Chen, M. I. Kennemer, K. Pothuraju, K. Konvicka, M. Tsoupko-Sitnikov, K. P. Pant, J. C. Ebert, G. B. Nilsen, J. Baccash, A. L. Halpern, G. M. Church, and R. Drmanac, "Accurate whole-genome sequencing and haplotyping from 10 to 20 human cells.," *Nature*, vol. 487, pp. 190–195, Jul 2012.

[25] T. IUM, "Encode project writes eulogy for junk dna," 2012.

[26] A. J. Iafrate, L. Feuk, M. N. Rivera, M. L. Listewnik, P. K. Donahoe, Y. Qi, S. W. Scherer, and C. Lee, "Detection of large-scale variation in the human genome.," *Nat Genet*, vol. 36, pp. 949–951, Sep 2004.

[27] J. Sebat, B. Lakshmi, J. Troge, J. Alexander, J. Young, P. Lundin, S. Månér, H. Massa, M. Walker, M. Chi, N. Navin, R. Lucito, J. Healy, J. Hicks, K. Ye, A. Reiner, T. C. Gilliam, B. Trask, N. Patterson, A. Zetterberg, and M. Wigler, "Large-scale copy number polymorphism in the human genome.," *Science*, vol. 305, pp. 525–528, Jul 2004.

[28] E. Tüzün, A. J. Sharp, J. A. Bailey, R. Kaul, V. A. Morrison, L. M. Pertz, E. Haugen, H. Hayden, D. Albertson, D. Pinkel, M. V. Olson, and E. E. Eichler, "Fine-scale structural variation of the human genome.," *Nat Genet*, vol. 37, pp. 727–732, Jul 2005.

[29] J. M. Kidd, G. M. Cooper, W. F. Donahue, H. S. Hayden, N. Sampas, T. Graves, N. Hansen, B. Teague, C. Alkan, F. Antonacci, E. Haugen, T. Zerr, N. A. Yamada, P. Tsang, T. L. Newman, E. Tüzün, Z. Cheng, H. M. Ebling, N. Tusneem, R. David, W. Gillett, K. A. Phelps, M. Weaver, D. Saranga, A. Brand, W. Tao, E. Gustafson, K. McKernan, L. Chen, M. Malig, J. D. Smith, J. M. Korn, S. A. McCarroll, D. A. Altshuler, D. A. Peiffer, M. Dorschner, J. Stamatoyannopoulos, D. Schwartz, D. A. Nickerson, J. C. Mullikin, R. K. Wilson, L. Bruhn, M. V. Olson, R. Kaul, D. R. Smith, and E. E. Eichler, "Mapping and sequencing of structural variation from eight human genomes.," *Nature*, vol. 453, pp. 56–64, May 2008.

[30] R. E. Mills, K. Walter, C. Stewart, R. E. Handsaker, K. Chen, C. Alkan, A. Abyzov, S. C. Yoon, K. Ye, R. K. Cheetham, A. Chinwalla, D. F. Conrad, Y. Fu, F. Grubert, I. Hajirasouliha, F. Hormozdiari, L. M. Iakoucheva, Z. Iqbal, S. Kang, J. M. Kidd, M. K. Konkel, J. Korn, E. Khurana, D. Kural, H. Y. K. Lam, J. Leng, R. Li, Y. Li, C.-Y. Lin, R. Luo, X. J. Mu, J. Nemesh, H. E. Peckham, T. Rausch, A. Scally, X. Shi, M. P. Stromberg, A. M. Stütz, A. E. Urban, J. A. Walker, J. Wu, Y. Zhang, Z. D. Zhang, M. A. Batzer, L. Ding, G. T. Marth, G. McVean, J. Sebat, M. Snyder, J. Wang, K. Ye, E. E. Eichler, M. B. Gerstein, M. E. Hurles, C. Lee, S. A. McCarroll, J. O. Korbel, and . G. Project, "Mapping copy number variation by population-scale genome sequencing.," *Nature*, vol. 470, pp. 59–65, Feb 2011.

[31] F. Antonacci, J. M. Kidd, T. Marques-Bonet, B. Teague, M. Ventura, S. Girirajan, C. Alkan, C. D. Campbell, L. Vives, M. Malig, *et al.*, "A large and complex structural polymorphism at 16p12. 1 underlies microdeletion disease risk," *Nature genetics*, vol. 42, no. 9, pp. 745–750, 2010.

[32] M. Ventura, C. R. Catacchio, C. Alkan, T. Marques-Bonet, S. Sajjadian, T. A. Graves, F. Hormozdiari, A. Navarro, M. Malig, C. Baker, C. Lee, E. H. Turner, L. Chen, J. M. Kidd, N. Archidiacono, J. Shendure, R. K. Wilson, and E. E. Eichler, "Gorilla genome structural variation reveals evolutionary parallelisms with chimpanzee.," *Genome Res*, vol. 21, pp. 1640–1649, Oct 2011.

[33] S. Yoon, Z. Xuan, V. Makarov, K. Ye, and J. Sebat, "Sensitive and accurate detection of copy number variants using read depth of coverage.," *Genome Res*, vol. 19, pp. 1586–1592, Sep 2009.

[34] C. Alkan, J. M. Kidd, T. Marques-Bonet, G. Aksay, F. Antonacci, F. Hormozdiari, J. O. Kitzman, C. Baker, M. Malig, O. Mutlu, S. C. Sahinalp, R. A. Gibbs, and E. E. Eichler, "Personalized copy number and segmental duplication maps using next-generation sequencing.," *Nat Genet*, vol. 41, pp. 1061–1067, Oct 2009.

[35] R. Redon, S. Ishikawa, K. R. Fitch, L. Feuk, G. H. Perry, T. D. Andrews, H. Fiegler, M. H. Shapero, A. R. Carson, W. Chen, E. K. Cho, S. Dallaire, J. L. Freeman, J. R. González, M. Gratacòs, J. Huang, D. Kalaitzopoulos, D. Komura, J. R. MacDonald, C. R. Marshall, R. Mei, L. Montgomery, K. Nishimura, K. Okamura, F. Shen,

M. J. Somerville, J. Tchinda, A. Valsesia, C. Woodwark, F. Yang, J. Zhang, T. Zerjal, J. Zhang, L. Armengol, D. F. Conrad, X. Estivill, C. Tyler-Smith, N. P. Carter, H. Aburatani, C. Lee, K. W. Jones, S. W. Scherer, and M. E. Hurles, "Global variation in copy number in the human genome," *Nature*, vol. 444, pp. 444–454, Nov 2006.

[36] D. F. Conrad, D. Pinto, R. Redon, L. Feuk, O. Gokcumen, Y. Zhang, J. Aerts, T. D. Andrews, C. Barnes, P. Campbell, T. Fitzgerald, M. Hu, C. H. Ihm, K. Kristiansson, D. G. Macarthur, J. R. Macdonald, I. Onyiah, A. W. C. Pang, S. Robson, K. Stirrups, A. Valsesia, K. Walter, J. Wei, W. T. C. C. Consortium, C. Tyler-Smith, N. P. Carter, C. Lee, S. W. Scherer, and M. E. Hurles, "Origins and functional impact of copy number variation in the human genome.," *Nature*, vol. 464, pp. 704–712, Apr 2010.

[37] S. A. McCarroll, T. N. Hadnott, G. H. Perry, P. C. Sabeti, M. C. Zody, J. C. Barrett, S. Dallaire, S. B. Gabriel, C. Lee, M. J. Daly, D. M. Altshuler, and I. H. C. , "Common deletion polymorphisms in the human genome.," *Nat Genet*, vol. 38, pp. 86–92, Jan 2006.

[38] J. O. Korbel, A. E. Urban, J. P. Affourtit, B. Godwin, F. Grubert, J. F. Simons, P. M. Kim, D. Palejev, N. J. Carriero, L. Du, B. E. Taillon, Z. Chen, A. Tanzer, A. C. E. Saunders, J. Chi, F. Yang, N. P. Carter, M. E. Hurles, S. M. Weissman, T. T. Harkins, M. B. Gerstein, M. Egholm, and M. Snyder, "Paired-end mapping reveals extensive structural variation in the human genome," *Science*, vol. 318, pp. 420–426, Oct 2007.

[39] K. R. Rosenbloom, J. Armstrong, G. P. Barber, J. Casper, H. Clawson, M. Diekhans, T. R. Dreszer, P. A. Fujita, L. Guruvadoo, M. Haeussler, *et al.*, "The ucsc genome browser database: 2015 update," *Nucleic acids research*, vol. 43, no. D1, pp. D670–D681, 2015.

[40] P. A. Fujita, B. Rhead, A. S. Zweig, A. S. Hinrichs, D. Karolchik, M. S. Cline, M. Goldman, G. P. Barber, H. Clawson, A. Coelho, *et al.*, "The ucsc genome browser database: update 2011," *Nucleic acids research*, p. gkq963, 2010.

[41] A. Martínez-Fundichely, S. Casillas, R. Egea, M. Ràmia, A. Barbadilla, L. Pantano, M. Puig, and M. Cáceres, "Invfest, a database integrating information of polymorphic inversions in the human genome," *Nucleic acids research*, pp. D1027–32, Jan 2014.

[42] A. Abyzov and M. Gerstein, "Age: defining breakpoints of genomic structural variants at single-nucleotide resolution, through optimal alignments with gap excision," *Bioinformatics*, vol. 27, no. 5, pp. 595–603, 2011.

[43] K. Chen, J. W. Wallis, M. D. McLellan, D. E. Larson, J. M. Kalicki, C. S. Pohl, S. D. McGrath, M. C. Wendl, Q. Zhang, D. P. Locke, X. Shi, R. S. Fulton, T. J. Ley, R. K. Wilson, L. Ding, and E. R. Mardis, "BreakDancer: an algorithm for high-resolution mapping of genomic structural variation.," *Nat Methods*, vol. 6, pp. 677–681, Sep 2009.

[44] R. P. Abo, E. P. Garcia, M. Ducar, R. Adusumilli, M. Breneiser, V. Rojas-Rudilla, L. M. Sholl, N. I. Lindeman, M. L. Meyerson, W. C. Hahn, *et al.*, "Breakmer: Detection of structural rearrangements in targeted next-generation sequencing data using kmers," *Cancer Research*, vol. 74, no. 19 Supplement, pp. 5321–5321, 2014.

[45] S. Suzuki, T. Yasuda, Y. Shiraishi, S. Miyano, and M. Nagasaki, "Clipcrop: a tool for detecting structural variations with single-base resolution using soft-clipping information," *BMC bioinformatics*, vol. 12, no. Suppl 14, p. S7, 2011.

[46] Z. Iqbal, M. Caccamo, I. Turner, P. Flicek, and G. McVean, "De novo assembly and genotyping of variants using colored de bruijn graphs.," *Nat Genet*, vol. 44, pp. 226–232, Feb 2012.

[47] J. Wang, C. G. Mullighan, J. Easton, S. Roberts, S. L. Heatley, J. Ma, M. C. Rusch, K. Chen, C. C. Harris, L. Ding, *et al.*, "Crest maps somatic structural variation in cancer genomes with base-pair resolution," *Nature methods*, vol. 8, no. 8, pp. 652–654, 2011.

[48] K. Trappe, A.-K. Emde, H.-C. Ehrlich, and K. Reinert, "Gustaf: Detecting and correctly classifying svs in the ngs twilight zone," *Bioinformatics*, p. btu431, 2014.

[49] J. Qi and F. Zhao, "ingap-sv: a novel scheme to identify and visualize structural variation from paired end mapping data," *Nucleic acids research*, vol. 39, no. suppl 2, pp. W567–W575, 2011.

[50] L. Yang, L. J. Luquette, N. Gehlenborg, R. Xi, P. S. Haseley, C.-H. Hsieh, C. Zhang, X. Ren, A. Protopopov, L. Chin, *et al.*, "Diverse mechanisms of somatic structural variations in human cancer genomes," *Cell*, vol. 153, no. 4, pp. 919–929, 2013.

[51] M. Mohiyuddin, J. C. Mu, J. Li, N. B. Asadi, M. B. Gerstein, A. Abyzov, W. H. Wong, and H. Y. Lam, "Metasv: an accurate and integrative structural-variant caller for next generation sequencing," *Bioinformatics*, p. btv204, 2015.

[52] J. Korbel, A. Abyzov, X. Mu, N. Carriero, P. Cayting, Z. Zhang, M. Snyder, and M. Gerstein, "PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data.," *Genome Biol*, vol. 10, p. R23, Feb 2009.

[53] Y. Jiang, Y. Wang, and M. Brudno, "Prism: pair-read informed split-read mapping for base-pair level detection of insertion, deletion and structural variants," *Bioinformatics*, vol. 28, no. 20, pp. 2576–2583, 2012.

[54] S. R. Landman, T. H. Hwang, K. A. Silverstein, Y. Li, S. M. Dehm, M. Steinbach, and V. Kumar, "Shear: sample heterogeneity estimation and assembly by reference," *BMC genomics*, vol. 15, no. 1, p. 84, 2014.

[55] Y. Li, H. Zheng, R. Luo, H. Wu, H. Zhu, R. Li, H. Cao, B. Wu, S. Huang, H. Shao, *et al.*, "Structural variation in two human genomes mapped at single-nucleotide resolution by whole genome de novo assembly," *Nature biotechnology*, vol. 29, no. 8, pp. 723–730, 2011.

[56] S. N. Hart, V. Sarangi, R. Moore, S. Baheti, J. D. Bhavsar, F. J. Couch, and J.-P. A. Kocher, "Softsearch: integration of multiple sequence features to identify breakpoints of structural variations," 2013.

[57] B. Zeitouni, V. Boeva, I. Janoueix-Lerosey, S. Loeillet, P. Legoix-Né, A. Nicolas, O. Delattre, and E. Barillot, "Svdetect: a tool to identify genomic structural variations from paired-end and mate-pair sequencing data," *Bioinformatics*, vol. 26, no. 15, pp. 1895–1896, 2010.

[58] M. Hayes, Y. S. Pyon, and J. Li, "A model-based clustering method for genomic structural variant prediction and genotyping using paired-end sequencing data," 2012.

[59] C. Lemaitre, L. Ciortuz, and P. Peterlongo, "Mapping-free and assembly-free discovery of inversion breakpoints from raw ngs reads," in *Algorithms for Computational Biology*, pp. 119–130, Springer, 2014.

[60] K. Chen, L. Chen, X. Fan, J. Wallis, L. Ding, and G. Weinstock, "TIGRA: a targeted iterative graph routing assembler for breakpoint assembly.," *Genome Res*, vol. 24, pp. 310–317, Feb 2014.

[61] F. Hormozdiari, I. Hajirasouliha, P. Dao, F. Hach, D. Yorukoglu, C. Alkan, E. E. Eichler, and S. C. Sahinalp, "Next-generation VariationHunter: combinatorial algorithms for transposon insertion discovery.," *Bioinformatics*, vol. 26, pp. i350–i357, Jun 2010.

[62] B. J. Trask, H. Massa, V. Brand-Arpon, K. Chan, C. Friedman, O. T. Nguyen, E. Eichler, G. van den Engh, S. Rouquier, H. Shizuya, *et al.*, "Large multi-chromosomal duplications encompass many members of the olfactory receptor gene family in the human genome," *Human molecular genetics*, vol. 7, no. 13, pp. 2007–2020, 1998.

[63] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows-Wheeler transform.," *Bioinformatics*, vol. 25, pp. 1754–1760, Jul 2009.

[64] H. Xin, D. Lee, F. Hormozdiari, S. Yedkar, O. Mutlu, and C. Alkan, "Accelerating read mapping with fasthash.," *BMC Genomics*, vol. 14 Suppl 1, p. S13, 2013.

[65] J. J. Smith, A. B. Stuart, T. Sauka-Spengler, S. W. Clifton, and C. T. Amemiya, "Development and analysis of a germline BAC resource for the sea lamprey, a vertebrate that undergoes substantial chromatin diminution.," *Chromosoma*, vol. 119, pp. 381–389, Aug 2010.

[66] A. Adey, H. G. Morrison, Asan, X. Xun, J. O. Kitzman, E. H. Turner, B. Stackhouse, A. P. MacKenzie, N. C. Caruccio, X. Zhang, and J. Shendure, "Rapid, low-input, low-bias construction of shotgun fragment libraries by high-density in vitro transposition.," *Genome Biol*, vol. 11, no. 12, p. R119, 2010.

[67] A. R. Quinlan and I. M. Hall, "Bedtools: a flexible suite of utilities for comparing genomic features," *Bioinformatics*, vol. 26, no. 6, pp. 841–842, 2010.

[68] F. Antonacci, J. M. Kidd, T. Marques-Bonet, M. Ventura, P. Siswara, Z. Jiang, and E. E. Eichler, "Characterization of six human disease-associated inversion polymorphisms.," *Hum Mol Genet*, vol. 18, pp. 2555–2566, Jul 2009.

[69] F. Antonacci, M. Y. Dennis, J. Huddleston, P. H. Sudmant, K. M. Steinberg, J. A. Rosenfeld, M. Miroballo, T. A. Graves, L. Vives, M. Malig, L. Denman, A. Raja, A. Stuart, J. Tang, B. Munson, L. G. Shaffer, C. T. Amemiya, R. K. Wilson, and E. E. Eichler, "Palindromic GOLGA8 core duplicons promote chromosome 15q13.3 microdeletion and evolutionary instability.," *Nat Genet*, vol. 46, pp. 1293–1302, Dec 2014.

[70] M. Fiume, E. J. Smith, A. Brook, D. Strbenac, B. Turner, A. M. Mezlini, M. D. Robinson, S. J. Wodak, and M. Brudno, "Savant genome browser 2: visualization and analysis for population-scale genomics," *Nucleic acids research*, vol. 40, no. W1, pp. W615–W621, 2012.

[71] D. W. Barnett, E. K. Garrison, A. R. Quinlan, M. P. Strömberg, and G. T. Marth, "Bamtools: a c++ api and toolkit for analyzing and managing bam files," *Bioinformatics*, vol. 27, no. 12, pp. 1691–1692, 2011.

Appendix A

Proofs

A.1 Inversion discovery probability

Assuming there is an inversion with breakpoints B1 and B2, the probability of picking a clone of length clone.length uniformly from the genome of length genome.length such that it will pass one of the breakpoints with at least the distance of one paired-end read which is PE.length from the breakpoint can be calculated as:

$$\begin{aligned} \text{P1} &= \text{P}(\text{clone.start} \in [\text{B1} - \text{clone.length} + \text{PE.length}, \text{B1} - \text{PE.length}]) \\ &= \frac{\text{clone.length} - 2\text{PE.length}}{\text{genome.length}} \end{aligned} \tag{A.1}$$

Now we give the probability of having a clone of size clone.length which is ideally obtained by a Gaussian distribution with mean of clone.μ and standard deviation of clone.σ. The probability is approximated from the truncated normal distribution.

$$\begin{aligned} \text{P2} = \text{P}(\text{clone}|\text{clone.length}) &= \int_{x=\text{clone.length}-1}^{\text{clone.length}} \frac{\frac{1}{\text{clone.}\sigma}\Phi\left(\frac{x-\text{clone.}\mu}{\text{clone.}\sigma}\right)}{\Phi\left(\frac{\text{genome.length}-\text{clone.}\mu}{\text{clone.}\sigma}\right) - \Phi\left(\frac{-\text{clone.}\mu}{\text{clone.}\sigma}\right)} \\ &= \frac{\int_{x=\text{clone.length}-1}^{\text{clone.length}} e^{\frac{-(x-\text{clone.}\mu)^2}{2}}}{\text{clone.}\sigma\left(e^{\frac{-(\text{genome.length}-\text{clone.}\mu)^2}{2}} - e^{\frac{-\text{clone.}\mu^2}{2}}\right)} \end{aligned} \tag{A.2}$$

Since S1 and S2 are independent, the probability of a clone passing B1 is:

$$\mathrm{P3} = \mathrm{P(S1|S2)} = \mathrm{P(S1)} \times \mathrm{P(S2)} \tag{A.3}$$

For all clones, we require at least one clone to cover B1, which means one occurrence of S3 in n times (Bernoulli). This probability is:

$$\mathrm{P4} = 1 - (1 - \mathrm{P3})^{\mathrm{n}} \tag{A.4}$$

where n is the number of clones and can be computed by:

$$\mathrm{n} = \frac{\mathrm{genome.length} \times \mathrm{physical.coverage}}{\mathrm{clone.}\mu} \tag{A.5}$$

Now we define P5 as the probability of having one clone covering B1 and another covering B2 when B2−B1+1 $\geq$ clone.length (given that B1 and B2 are far enough from each other) is:

$$\mathrm{P5} = \mathrm{P4(n)} \times \mathrm{P4(n-1)} = \left(1 - (1 - \mathrm{P3})^{\mathrm{n}}\right).\left(1 - (1 - \mathrm{P3})^{\mathrm{n-1}}\right) \tag{A.6}$$

Now we should calculate the probability of having two clones in the same pool. Assuming that the procedure of picking clones is independent from each other and the distribution is uniform:

$$\mathrm{P6} = \mathrm{P}(\mathrm{clone_i} \in \mathrm{pool_k}\ \&\ \mathrm{clone_j} \in \mathrm{pool_k}) = \frac{1}{\mathrm{pools.count}^2} \tag{A.7}$$

Finally we can define the probability of a *discoverable inversion* which means there is a clone passing B1 and another passing B2 while these two clones do not overlap:

$$\mathrm{P7} = \mathrm{P(findable\ inversion)} = \mathrm{P5} \times \mathrm{P6} = \frac{\left(1 - (1 - \mathrm{P3})^{\mathrm{n}}\right).\left(1 - (1 - \mathrm{P3})^{\mathrm{n-1}}\right)}{\mathrm{pools.count}^2} \tag{A.8}$$

Here we have the probability of having clones such that a given inversion is discoverable.

A.2 The set cover approximation problem

Initially we tried to formulate the split clone clustering problem as a set cover problem, similar to the approach use by VariationHunter [14]. However in most cases we observed

that the set cover approximation returns the inversion with one breakpoint precisely, while the other breakpoint is far from the exact locus. The problem is due to the nature of inversions where the breakpoints are located on duplications and highly repeated regions. For this reason, the inversion signatures, both split clones and read pairs, will have almost complete cliques for each inversion with many edges between the neighboring cliques. The equivalent for such a situation with a set cover formulation will be neighboring sets sharing the some elements as shown in Equation 9 and Appendix Figure A.1 :

$$\begin{aligned} U &= \{A, B, C, D, E, F, G, H, I, J\} \\ S &= \{\{A, B, C, D, H\}, \{C, D, E, F, H\}, \{F, G, H\}, \{H, I, J\}\} \end{aligned} \tag{A.9}$$

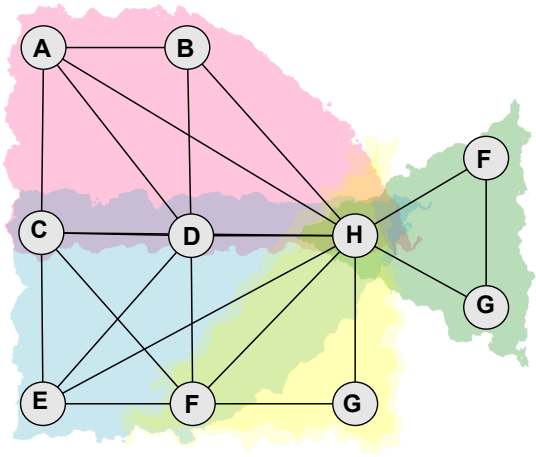


Figure A.1: Graph representation for the example given in Equation A.9. Each colored area represents a clique (equivalent to a set in the set cover formulation).

The set cover performing in any order will fail to recognize the most reliable breakpoint set because its greedy approach just chooses the set with the highest number of *new elements* which might lead to disfavoring other sets as their elements will become *found already*, and as a result, it will get stuck in a local optimum which is most likely the duplications near the breakpoint rather than the actual inversion itself. In contrast, if we choose a maximal quasi-clique approach, it can jump over these in-between-clique-edges and find the actual inversion. The effectiveness of the quasi-clique approach was observed on the second simulated data set. The problem of set cover approximation can

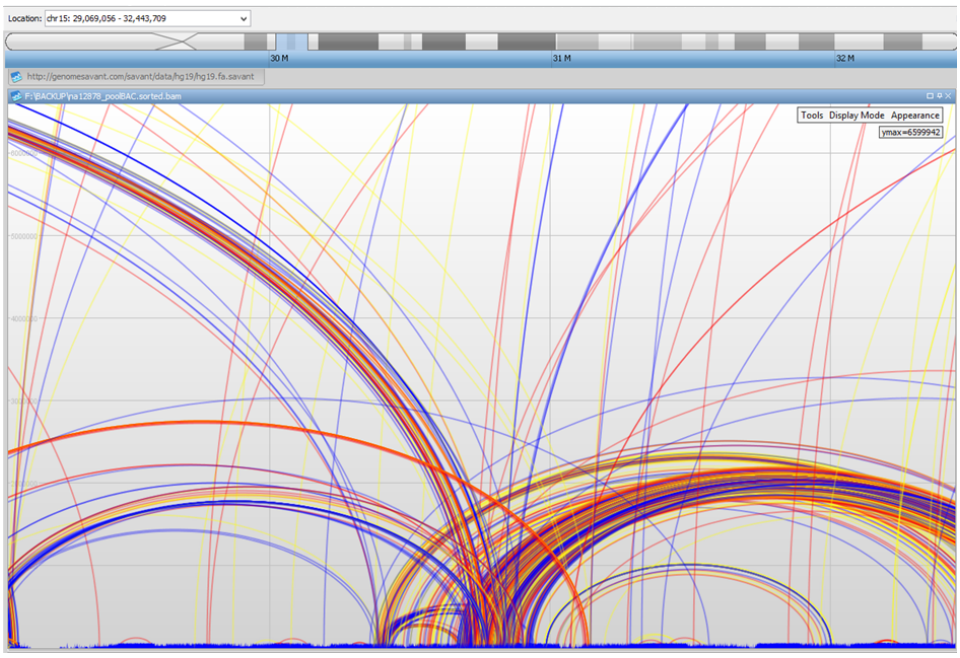


Figure A.2: Mapped paired-end reads around the HsInv1049 inversion of the NA12878 individual illustrated by SAVANT.

Red arcs display the discordant length mapping paired-end reads, dark blue represent the one read inverted, yellow arcs represent the everted paired-end reads and the lighter blue forward, reverse, or concordant length paired-end reads. Reads were mapped by BWA in this example.

be solved to some extend by applying a semi-randomization technique in compare to ordered set cover approximation, but this will cause the approximation rate to be unpredictable, and therefore, unreliable. The following SAVANT [70] figure shows a real example of such in-between-clique-edges. Notice the humps made by the paired-end reads mapping around the HsInv1049 inversion of the NA12878 individual with breakpoint 1 at chr15:30,370,112−30,910,305 and breakpoint 2 at chr15:32,445,408−32,899,708. The quasi-cliques around the original inversion clique can be seen clearly in this picture.

A.3 Clone overlap probability

We tried to evaluate the probability of clone overlap as some inferred clones of size larger than expected were observed which we suspected them to be due to overlaps in some

pools. The computational complexity of calculating the exact probability of overlap in a given set pool is too expensive ($O(n^{nm})$ where n is the number of clones and m is the length of the genome). In the real data of NA12878, there are approximately 230, 389, and 153 clones in each pool of set 1, 2, and 3, respectively. To evaluate the probability of clone overlap for each cutoff (number of clones overlapping), for maximum $2^{63} - 1$ test cases we extensively simulated a number of random clones in 288 pools (from normal distribution of μ=137 Kbp and σ=40 Kbp with cutoff 125 Kbp and 175 Kbp) and counted the average number of times there were x overlaps (for x=1 to total number of clones). Each test was stopped when the average number became stable to the thousands for 1000 consequent runs. This was repeated 1000 times and averaged for each cutoff (number of clones overlapping). The results are presented in the following tables. Figures are obtained by `RapidMiner`[1].

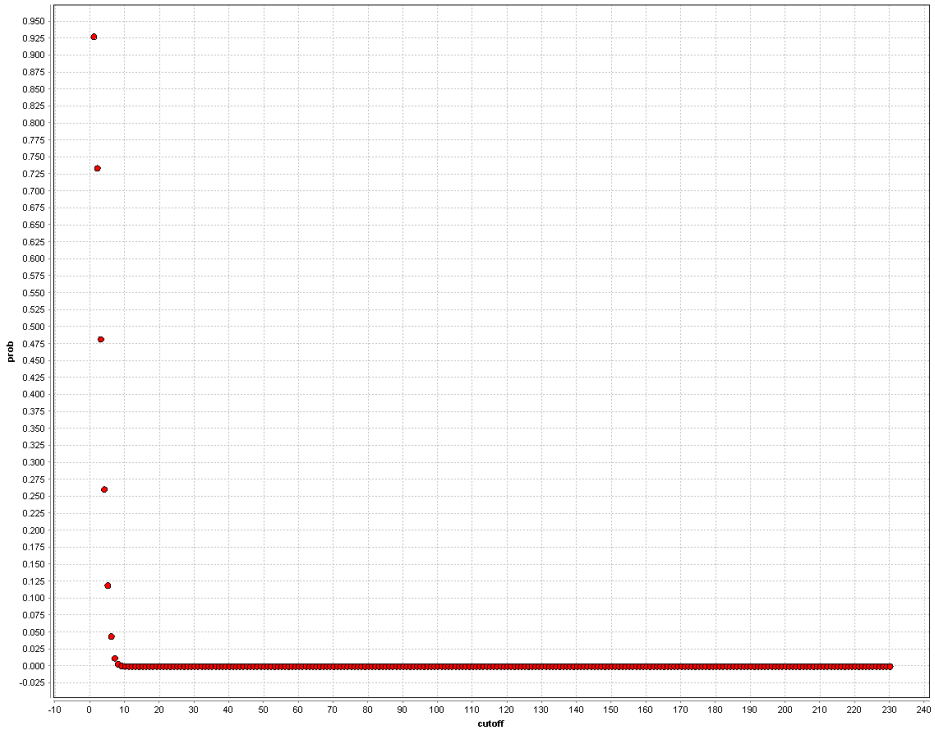


Figure A.3: Probability of overlapping for each number of clones estimated for set1 of pooled clone data of NA12878 with 230 clones per pool.

[1]https://rapidminer.com/

Table A.1: Exact values of overlapping probabilities estimated for set 1 of pooled clone data of NA12878 with 230 clones per pool.

cutoff	**prob**
1	92.789%
2	73.407%
3	48.212%
4	26.082%
5	11.916%
6	4.412%
7	1.210%
8	0.315%
9	0.072%
10	0.021%
11	0.009%
12	0.001%
13-230	0.00%

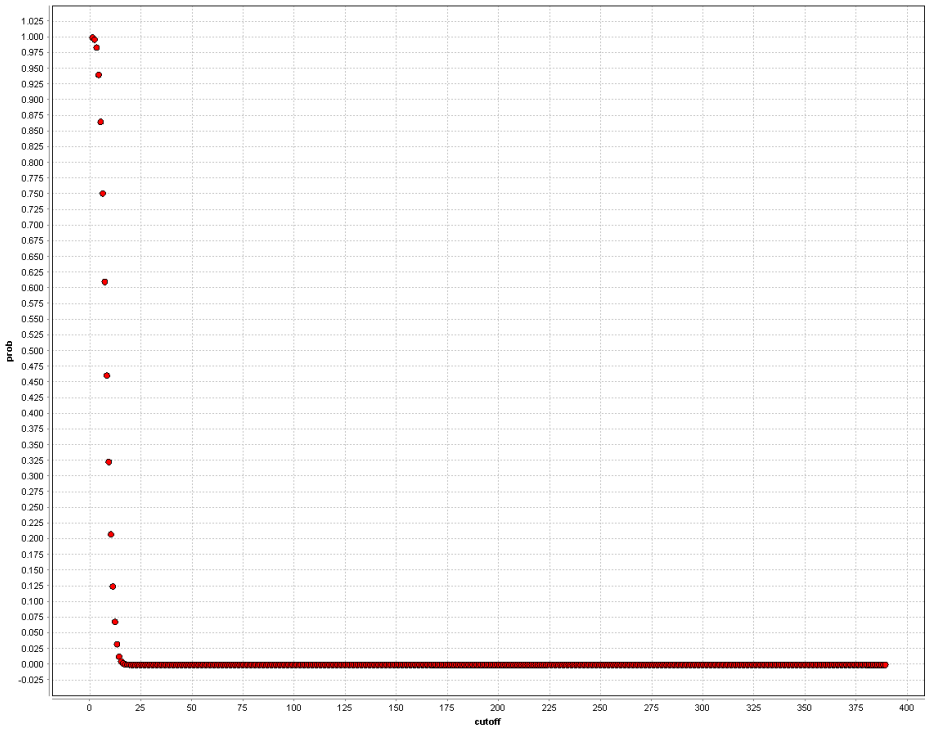


Figure A.4: Probability of overlapping for each number of clones estimated for set2 of pooled clone data of NA12878 with 389 clones per pool.

Table A.2: Exact values of of overlapping probabilities estimated for set2 of pooled clone data of NA12878 with 389 clones per pool.

cutoff	**prob**
1	99.967%
2	99.669%
3	98.380%
4	94.057%
5	86.551%
6	75.149%
7	61.036%
8	46.090%
9	32.326%
10	20.777%
11	12.479%
12	6.847%
13	3.280%
14	1.306%
15	0.549%
16	0.240%
17	0.075%
18	0.025%
19	0.011%
20	0.004%
21	0.001%
22	0.001%
23-389	0.00%

Table A.3: Exact values of of overlapping probabilities estimated for set3 of pooled clone data of NA12878 with 153 clones per pool.

cutoff	**prob**
1	68.498%
2	31.823%
3	10.719%
4	2.403%
5	0.436%
6	0.072%
7	0.013%
8	0.001%
9-153	0.00%

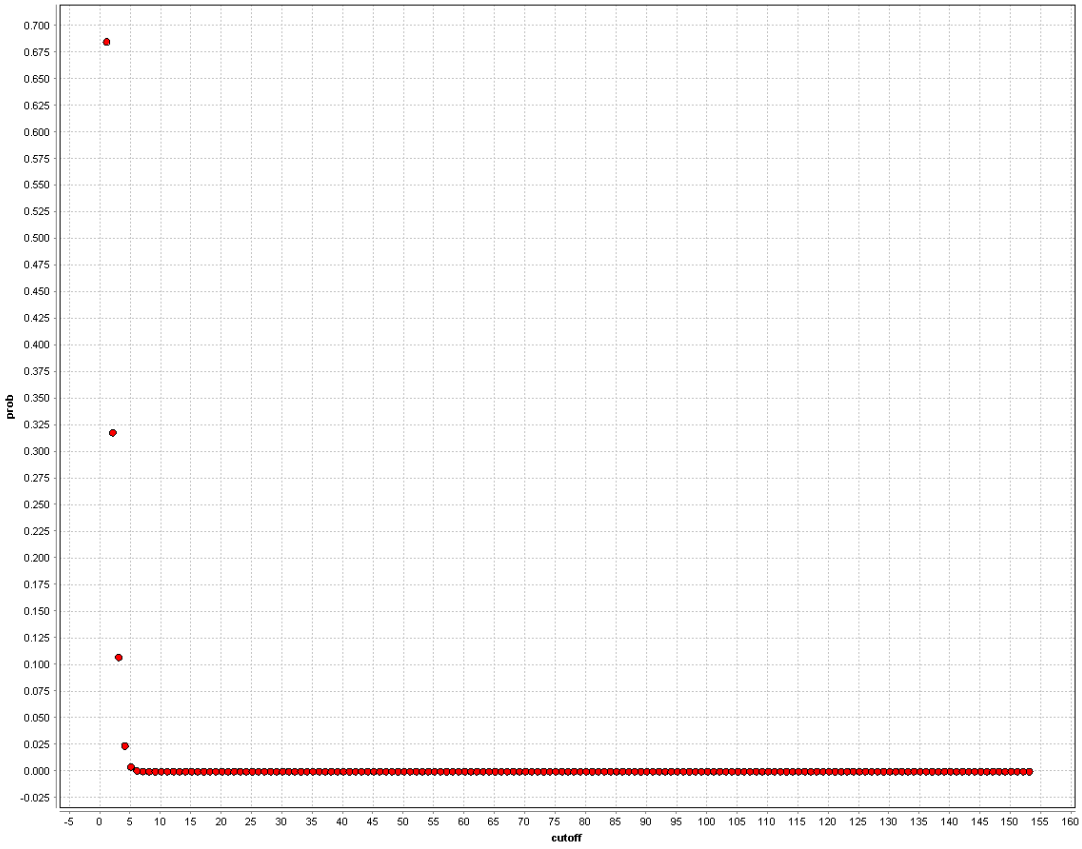


Figure A.5: Probability of overlapping for each number of clones estimated for set3 of pooled clone data of NA12878 with 153 clones per pool.

Appendix B

Parameter adaption

B.1 Clone reconstruction parameters

To reconstruct the clones from the normally mapping paired-end reads of each pool, we first look for windows of a minimum size which is covered by paired-end fragments by a pre-defined coverage rate. These well-covered windows are called clone seeds and are further extended to any existing fragment to the left or right at a given distance. In order to evaluate the best parameters for the minimum clone seed size, minimum coverage, and extension distance we applied a grid optimization on simulated data. Random clones on chromosome 1 with normally distributed sizes of (μ=150 Kbp , σ=40 Kbp) in 288 pools at 3X physical coverage were simulated and then fragmented with `wgsim` [1] at 3X, 5X, 10X, 15X, and 20X sequencing coverage for BWA, and 3X and 5X for mrFAST with size (μ=600 bp,σ=60 bp) and mapped back to reference chromosome 1 using BWA [63] and mrFAST, respectively. The parameter grid used is given in Table B.1. Due to duplicated regions and gaps and overlapping clones, not all clones can be precisely retrieved. The maximum rate of clone reconstruction requiring at least 90% reciprocal intersection is given in Table 4.2. It is worth mentioning that the reconstruction rate did not improve by increasing the coverage to 15X and 20X. Also, contrary to our expectation, mrFAST

[1] https://github.com/lh3/wgsim

aligner could not perform as precisely as the BWA aligner. The optimum set of parameters were minimum clone seed length of 6.5 Kbp, minimum coverage of 50%, and extension distance of 1500 bp. However in the case of real data where split clones occur, the window size should be set to the maximum fragment size such to not miss any split clone smaller than the window size. As it can be observed dipSeq relies on sufficient physical coverage (i.e. clones per pool) rather than sequencing coverage and can perform precisely in low sequence coverages.

Table B.1: Grid for parameter optimization for clone reconstruction.

parameter	**min**	**max**	**step size**	**number of steps**
min seed length	3,000	146,000	500 up to 10,000 1,000 afterwards	160
min coverage	0.5	1.0	0.1	5
read extension distance	1,000	10,000	1,000	10
			total	8,000

B.2 Parameter optimization of the maximal quasi-clique

In order to find the optimum parameters for the maximal quasi-clique approximation algorithm proposed by [22], 100 random graphs each including 4 highly connected quasi-cliques were produced and on each, a grid optimization was applied. The graphs are not randomly expected cases, but rather worse case scenarios that might occur and more similar to what we have observed in the real data set; meaning the neighboring nodes are connected with a higher probability and there are many connections between the hidden quasi-cliques and also, there exists many missing edges within each quasi-clique. The algorithm used to optimize the parameters for the maximal quasi-clique approximation is given in Algorithm 1.

It was observed that tabu ≤ 5 results into instability and slow convergence while values $\gg 10$ result in poor performance. Thus, dipSeq sets the tabu relative to the size of the nodes of the graph ($log(n)$). Also, for small number of nodes (< 100) high lambda and gamma performed better, but as the number of nodes increased and the quasi-cliques

Algorithm 1 Quasi Clique Parameter Optimization

1: **procedure** OPTIMIZEQUASICLIQUEPARAMS
2: for case $\leftarrow$ 1 to 99
3: $G \leftarrow$ a new graph
4: $Sets[1..5] \leftarrow$ make 4 sets of nodes each of random size $[4 \times 2^{\lfloor \frac{case}{10} \rfloor}, 6 \times 2^{\lfloor \frac{case}{10} \rfloor}]$
5: $n \leftarrow |set1| + |set2| + |set3| + |set4|$
6: place all the nodes in G in order of the set and label them from 1 to n
7: add another random $[4 \times 2^{\lfloor \frac{case}{10} \rfloor}, 6 \times 2^{\lfloor \frac{case}{10} \rfloor}]$ nodes in between the nodes of G
8: $\forall i, j \in G.nodes$ add $edge(i, j)$
with a probability of $\begin{cases} 80\%, & \text{if } (node_i \& nodej \in the\ same\ set) \\ distance^{-2} \times 60\%, & \text{otherwise} \end{cases}$
where distance is the difference of the order of the two nodes in the graph
9: for each $tabu \in \{1, 2, ..., case/2\}$ & $lambda \in \{0.1, 0.2, ..., 0.9\}$ & $gamma \in \{0.1, 0.2, ..., 0.9\}$
$Solution \leftarrow MaximalQuasiClique(G, tabu, lambda, gamma)$
$Score[case, n, tabu, lambda, gamma]$ =((number of real cliques) − (number of cliques found in Solution)
+ (number of elements in each clique that were found)) / $\lfloor \frac{case}{10} \rfloor$
10: find the highest scoring point of (n, tabu, lambda, gamma)

* Note that no penalty is applied if the algorithm returns the set with additional nodes because this will not cause any difference in the final inversion detection.

overlapped more, increasing the lambda and gamma caused the algorithm to return only the largest quasi-clique with the most connected nodes of that clique. For larger graph sizes, lambda and gamma close to 0.5 performed better. Observing that the cliques in the first simulation data (physical coverage 3-4X) have hundred of nodes where lambda and gamma near 0.5 held the highest scores in that range, in the next phase, we ran the algorithm on simulation 1 data set (see section 1.9) on the inferred clones from BWA mapped read pairs with 10X coverage for a grid of lambda$\in\{0.4,0.5,0.6\}$ and gamma$\in\{0.4,0.5,0.6\}$. As a result the optimum values for lambda and gamma were 0.5 and 0.6, respectively.

Appendix C

Code

Implementation of the dipSeq algorithm is available at https://github.com/BilkentCompGen/dipseq

Appendix D

Data

After simulations, dipSeq was applied to the pooled clone data from the genome of the NA12878 individual. Some statistics on the data are given bellow.

Table D.1: Number and percentage of mapping paired-end reads before and after removing duplicated ones

Set	Before	After	Distinct	Duplicated
set1	382,782,082	324,302,909	84.72%	15.28%
set2	223,707,355	190,888,484	85.33%	14.67%
set3	420,380,434	383,907,969	91.32%	8.68%
ALL	102,686,9871	899,099,362	87.56%	12.44%

Table D.2: Average number of normal size clones (125 Kbp-175 Kbp) inferred for each pool in each set vs. the expected number of clones

Clones	set1	set2	set3
Expected	230	389	153
With 0s	162.22	238	75.75
Without 0s	179	304.64	151.5

With 0 included the pools that had no inferred clones at all. Assuming that those probes might have been problematic, we also give the average numbers without including pools with zero clones as *Without 0*. The difference is due to error or split clones.

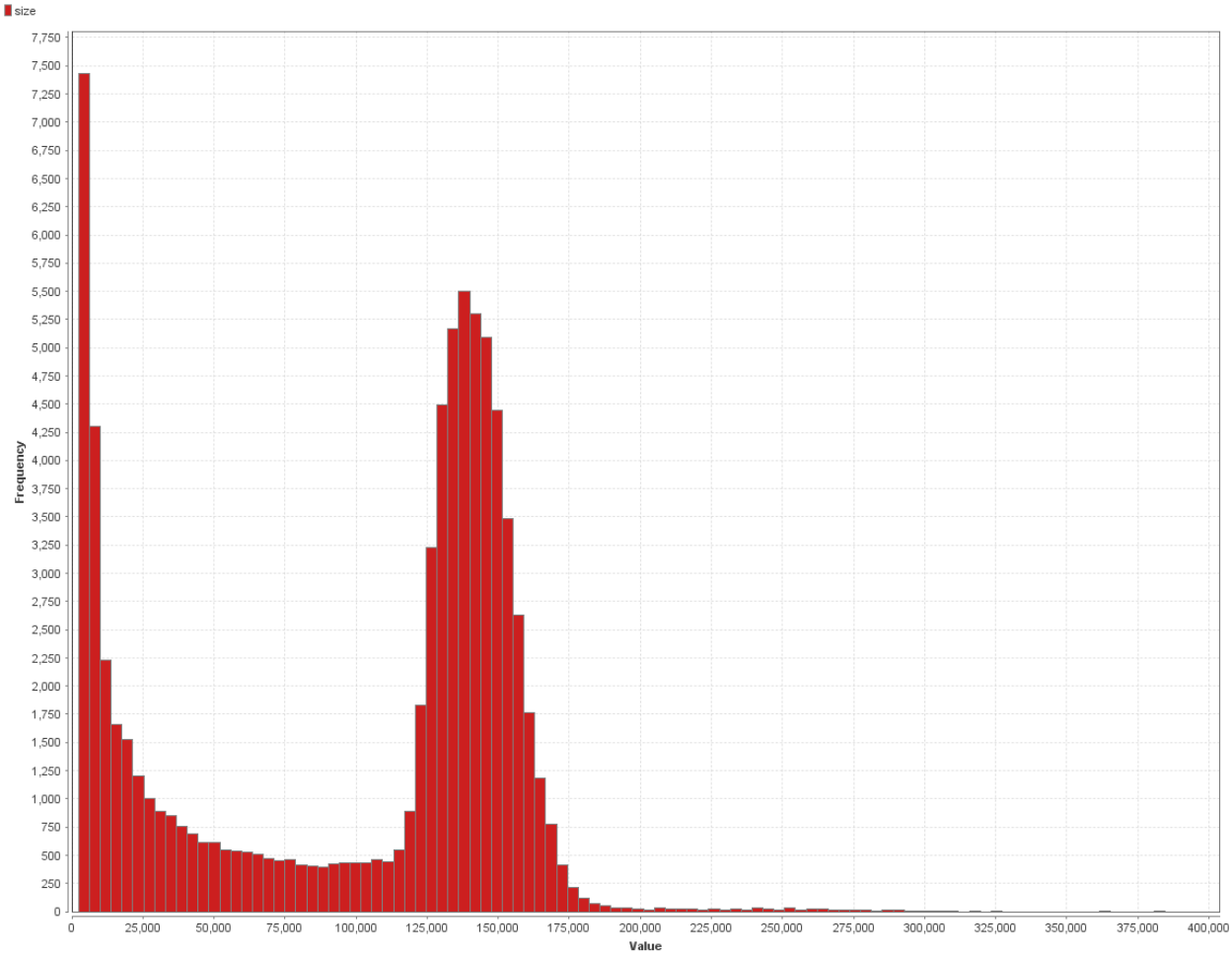


Figure D.1: Histogram of inferred clone size with 100 bins

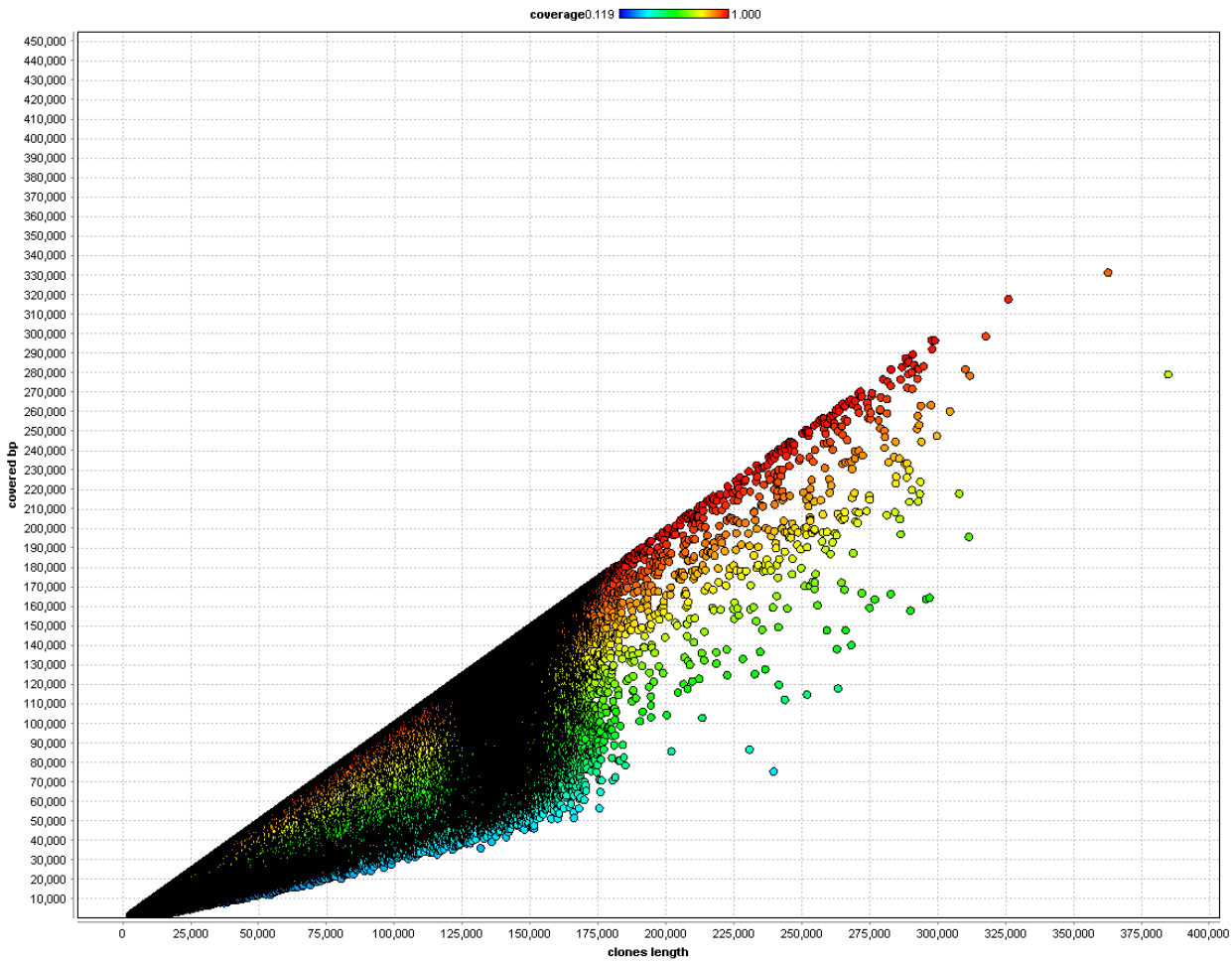


Figure D.2: Scatter plot of covered bp over clone size colored by coverage rate: It can be observed that clones of average size or larger are better covered

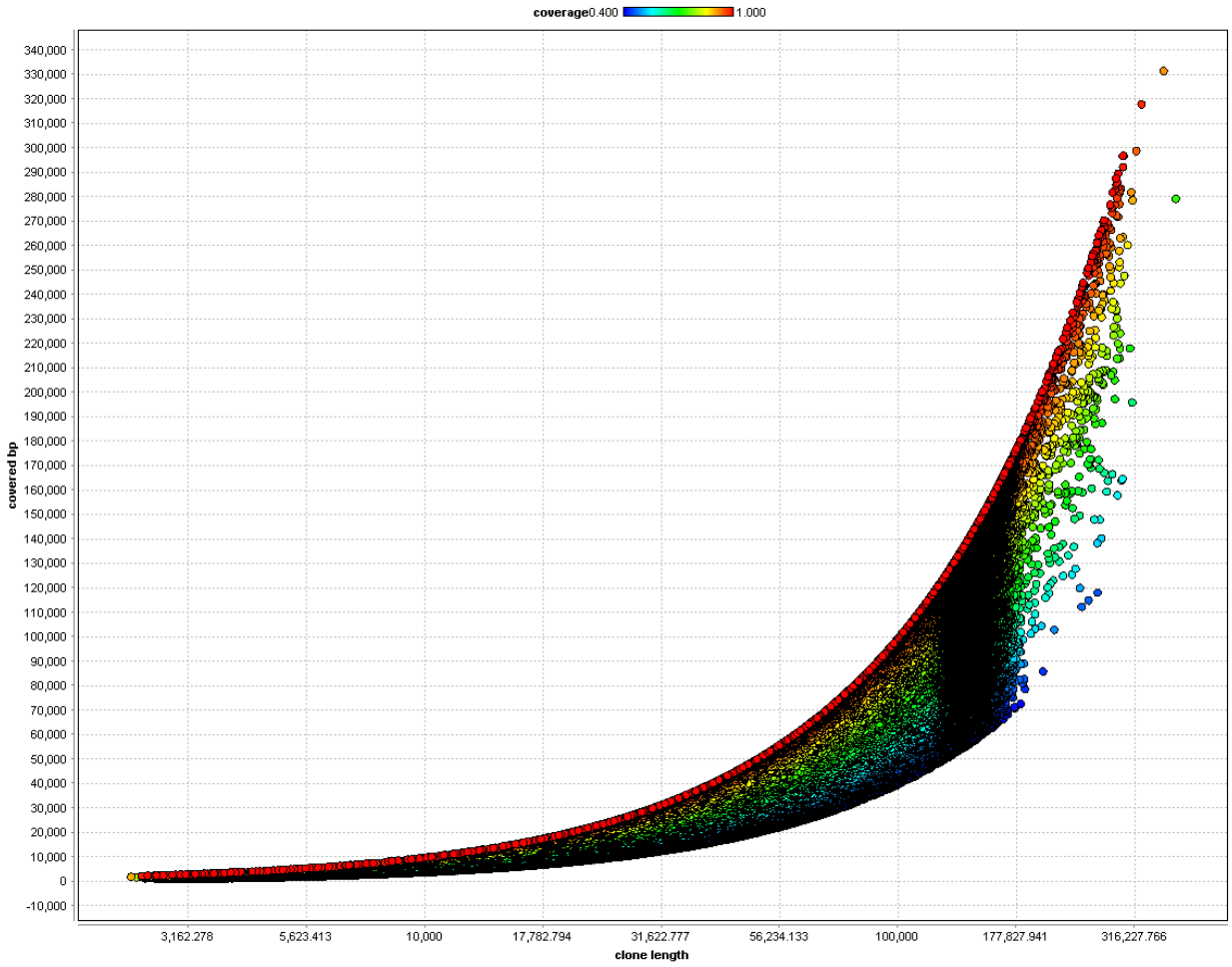


Figure D.3: Scatter plot of covered bp over log of clone length colored by coverage rate with cutoff of 40% coverage: It can be observed that clones of average size or larger are better covered

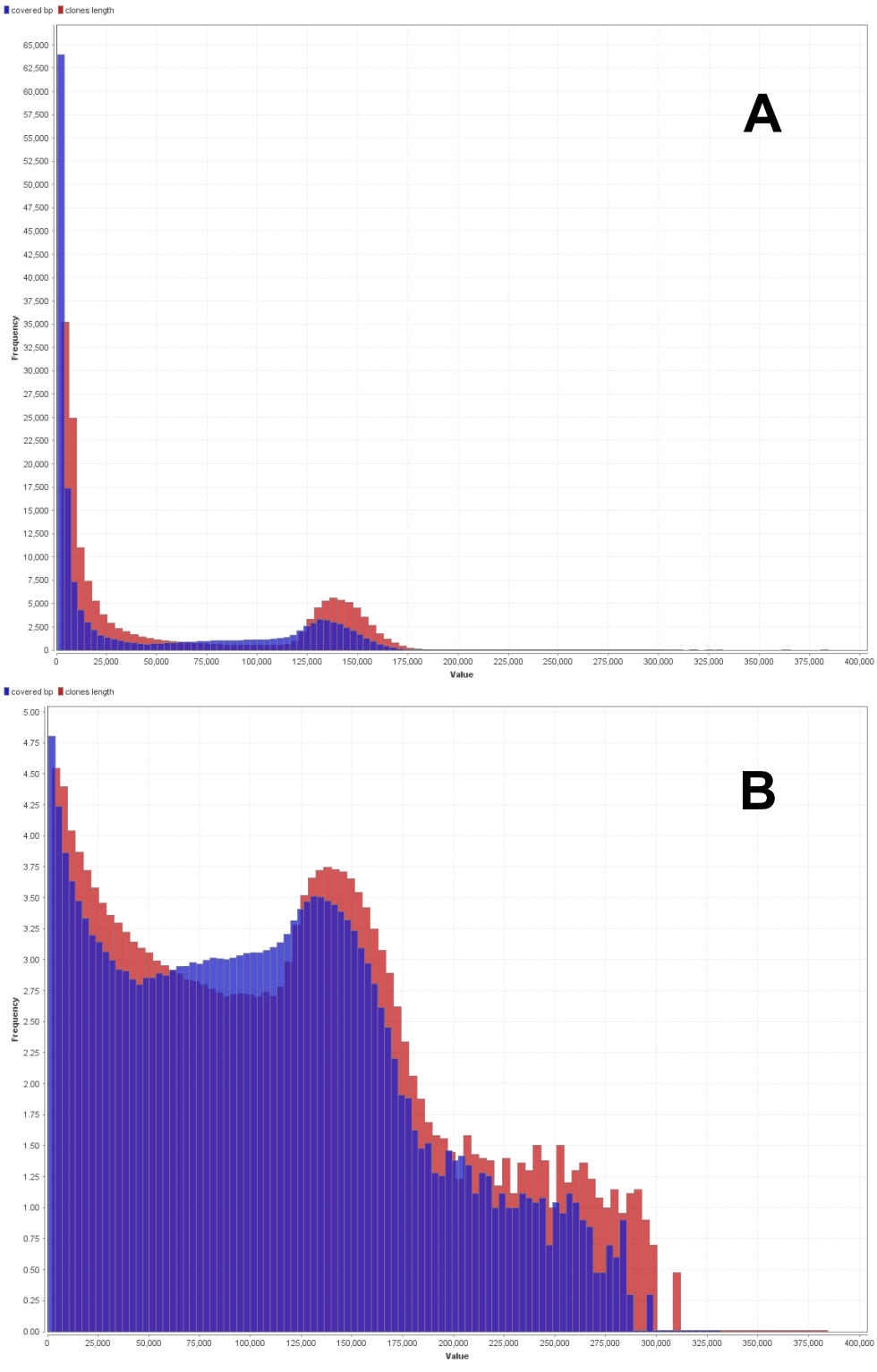


Figure D.4: (A) Histogram of covered bp over clone length with 100 bins and (B) Histogram of log of covered bp over log of clone length with 100 bins