NEW ALGORITHMS FOR STRUCTURAL VARIATION DETECTION BY DE NOVO GENOME ASSEMBLY

(Thesis format: Monograph)

by

Ehsan Haghshenas

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Masters of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

Structural variations (SVs) are changes in the human genome that are reported in several studies to be associated with some diseases. Therefore, designing methods to find these types of variations would help us for early detection of those diseases and utilizing new treatment methods such as personalized medicine. Currently computational methods are applied to find structural variations from short reads obtained by Next Generation Sequencing (NGS) platforms. Usually each method has more power in finding particular types or sizes of SVs and limitations in finding others. Thus, still new approaches and methods are on demand for SV discovery.

In this thesis, we introduce two new methods based on a de novo assembly framework called SAGE for detecting SVs. We compare our proposed methods with existing ones which are based on the same approaches. This comparison shows that our methods are able to detect more SVs from the validation sets (true SVs) than the compared methods.

Keywords: structural variation, copy number variation, de novo assembly, next-generation sequencing

To my loving family, especially Mom and Dad

Acknowlegements

First and foremost, I would like to express my sincerest gratitude to my supervisors, Dr. Lucian Ilie and Dr. Roberto Solis-Oba for their continuous support throughout my program. In addition to their guidance and encouragement during this project, I learned from them how a good researcher should think and work, which I am sure will help me in my future life. I also appreciate their patience in reading and revising my thesis.

I am grateful to my dear friends Ehsan and Yiwei for all the fun we had together. I am very thankful to the best friends I made in London, Samar and Meysam for all of their invaluable helps during this time. I would also like to thank my great friends Parisa, Shervin, Navid and Kaveh who helped me not feel alone far from home.

last but not least, I thank my family members for their non-stop supports that always keep me motivated.

Contents

List of Figures

List of Tables

Acronyms

DNA	Deoxyribonucleic Acid
bp	base pair(s)
NGS	Next Generation Sequencing
HTS	High Throughput Sequencing
SNP	Single Nucleotide Polymorphism
SNV	Single Nucleotide Variation
SV	Structural Variation
CNV	Copy Number Variation
aCGH	array Comparative Genome Hybridization
EWT	Event-Wise Testing
MCF	Minimum Cost Flow
GSV	Genome Structural Variation Consortium database
FDR	False Detection Rate

Chapter 1

Introduction

Every scientist in the field of Biology accepts that a better understanding of the human genome will help to get a better understanding of ourselves. This is due to the undeniable relationship between the human genome and all its biological functions. During the past few decades this understanding has been improved with non-stop advancements in the field of Deoxyribonucleic Acid (DNA) sequencing.

Indeed Next Generation Sequencing (NGS) technologies have been an important breakthrough in DNA sequencing and in general in biological sciences. By introducing these technologies and obtaining a clearer view of the human genome in the past decade, new problems have arisen that solving each one of them will help us in our path of gaining a comprehensive understanding of the human genome structure.

Structural variation (SV) detection and Copy Number Variation (CNV) detection are examples of such problems that have attracted a lot of attention during the past few years. The reason is that variations (differences in two genomes) are sources of all of the differences between two organisms. On the other hand, many SVs are reported to be associated with diseases. Therefore, obtaining a comprehensive map of SVs in the human genome will help design new treatments such as personalized medicine.

SV detection is a difficult problem, not only because SVs vary in size widely and there are many types of them, but also due to the fact that they mostly occur within repetitive regions of DNA [3]. Early methods for SV detection were experimental methods which had several limitations and could not detect all types of SVs. As a result, they

were replaced by computational methods that are more powerful. There are four main approaches for computational SV detection, and each of them has advantages and limitations. It is well known that computational SV detection methods are complementary and different methods should be used to find a good map of structural variations.

In this thesis two new algorithms are proposed for SV detection, both of them based on a de novo assembly framework called SAGE. The first algorithm is called SAGEcnv and it employs the new idea of utilizing estimated copy count of short reads to detect Copy Number Variations (CNVs). The second method is called SAGEsv and it tries to improve the assemblies produced by SAGE in order to obtain better assemblies and use them to detect SVs through a comparison with the reference genome.

1.1 Thesis overview

The remaining of this thesis is organized as follows. In Chapter 2 we present the necessary background on human genome and structural variations as well as previous work in structural variation detection. We briefly explain the main approaches and tools in the state of the art and mention advantages and disadvantages of each one of them.

Then we explain our proposed methods. SAGEcnv is described in Chapter 3. Since our proposed methods are both based on the SAGE assembler, we give a concise introduction to this framework at the beginning of this chapter as well. Chapter 4 describes SAGEsv and shows how the use of more information from the reference genome helps us to improve assemblies.

The results and a discussion about them are presented in Chapter 5. Finally, Chapter 6 gives our conclusions and possible future research based on this work.

Chapter 2

Genomic Variations

2.1 Molecular biology primer

2.1.1 DNA

Deoxyribonucleic Acid (DNA) is a complex molecule within living organisms' cells, composed of four types of *nucleotides* known as *adenine*, *cytosine*, *guanine* and *thymine*. It is simpler to imagine DNA as a long string of four letters: A, C, G and T, each of which represents one type of DNA nucleotides, respectively. Genomes in complex organisms are formed by a number of DNA molecules called *chromosomes*.

Nucleotides of a DNA string are attached together using a connection called *phosphodiester bond*. However, this is not the only connection in the DNA structure. DNA has a double helix structure which is made of two sequences of nucleotides called strands (see Figure 2.1). Each nucleotide in one strand is attached to one nucleotide in the opposite strand through a chemical connection called *hydrogen bond* and forms a unit called *base pair* (bp). The characteristics of DNA nucleotides are such that force an adenin (A) to be always paired with a thymine (T), and a cytosine (C) to be always paired with a guanine (G).

Hence, if we suppose that an A base is complementary of a T base and a C base is complementary of a G base, we conclude that one DNA strand is the reverse complement of the other strand. So from the computational point of view, we can consider only one

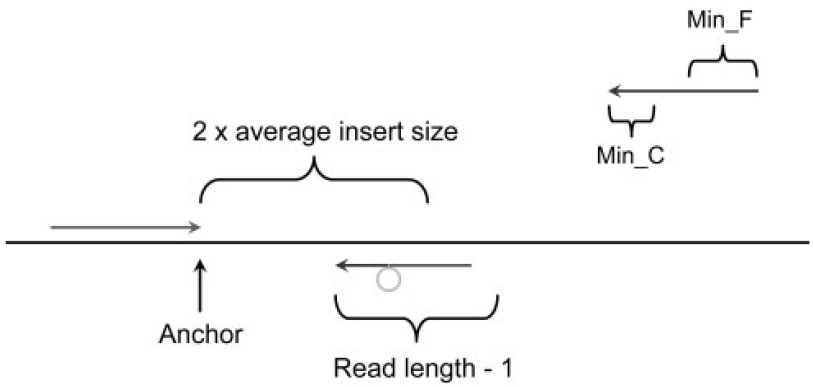

Figure 2.11: Detection of short insertion using Pindel. For insertion, searching for minimum and maximum unique substrings from the 3' end of the unmapped read is done within the range of two times of the insert size starting from the anchor point. The search range for minimum and maximum unique substrings from the 5' end of unmapped read is within read length – 1 starting from the already mappd 3' end of the unmapped read. In this the whole read cannot be reconstructed certainly and the extra bases are an inserted fragment compared to the reference genome [79].

two ends of the paired-end read.

SOAPsv

SOAPsv is a tool based on sequence assembly approach which uses assembled scaffolds obtained by SOAPdenovo [44]. This method is able to detect structural variants of small and intermediate size (1 bp-50 Kbp) and their precise breakpoints [46]. To use this method, first the short read dataset should be used to assemble scaffolds using de novo sequence assembly algorithm, SOAPdenovo. Details of the SOAPdenovo algorithm can be found in [44].

Then, assembled scaffolds are aligned to the reference genome using BLAT [32]. BLAT is designed to find alignments very quickly instead of very carefully. This gives a big picture of possible alignments of a scaffold to a specific chromosome. Next, scaffolds which are aligned to the same chromosome are chosen for more careful alignment to that chromosome. This is done by the LASTZ alignment tool [23] with the ambiguous 'N' treatment and gap-free extension tolerance up to 50 Kbp options enabled. Scaffolds with no hit using BLAT, are aligned to the whole reference genome using LASTZ with the

same options.

In the next step, the best hit of every single location on each chromosome is chosen using a utility called axtBest [63]. Then these hits are analyzed to find putative SV calls. If a gap is opened in the reference, it shows existence of an insertion and if a gap is opened in the scaffold a deletion could be found.

The last step is to validate putative SV calls and report the validated SV calls as detected. SOAPsv uses different approaches for validating small SVs and larger SVs. For SVs $\leq$50 bp, **short reads** are mapped to the reference genome using BWA [42]. If at least four reads have alignments that support a candidate SV call (position and length of gap opening should be in concordance with the prediction) then this SV call is validated and reported.

For SVs >50 bp, a measure called *S/P ratio* is defined. To calculate S/P ratio, short reads are aligned to the reference genome (both as single-end and paired-end reads). For a candidate SV, S/P is calculated as the ratio between the number of normally mapped paired-ends reads (which have the expected orientation and insert size) and single-end aligned reads (which are originally from a paired-end read but only one end is aligned) in a 50 bp flanking region of the SV boundaries. Then a Fisher's exact test is performed to test whether the S/P ratio of each structural variation is significantly different from the S/P ratio of the whole genome.

CNVer

CNVer is a tool for CNV detection that takes advantage of both read depth and read pair approaches [52]. This is done by combining read depth and read pair signatures with the reference genome in a novel framework called *donor graph*. Then the problem of CNV detection is modelled as a minimum cost flow problem on this donor graph. By solving this problem, in addition to SV calling, CNVer is able to predict the copy count of each region.

This method has several advantages over methods which use only one of the read depth or read pair signatures. First, it is capable of identifying the breakpoints of deletions (detected by read depth signatures) using the read pair signatures. Moreover, it can

detect insertions larger than the insert size, which is not possible using only read pair signatures. Furthermore, using two signatures yields a more robust framework that indicates a reduction in the false positive rate.

However, the authors stated that there is still a fraction of CNVs found by other methods like RDxplorer that CNVer is not able to find (even using both read depth and read pair signatures). Same thing can be observed by almost all other SV/CNV detection methods. In fact, it is well known that current approaches and tools for SV/CNV detection are complementary [3].

Other methods

We explained only a few of the available methods for SV/CNV detection that are utilized in the 1000 genome project (except CNVer). We present below a list of methods for each of the approaches mentioned above.

Read depth approach : There are many read depth approach methods among which the bast are RDxplorer [80], CNVnator [2], JointSLM [48], CNVem [72]. In addition, there are a few tools that take advantage of a control genome. They are usually able to find changes in a tumour genome (donor genome) relative to a normal genome (control genome). Segseq [9], CNVseq [77], CNAseq [31], rSW-seq [35], BICseq [75] and seqCBS [65] are examples of such methods.

Read pair approach : Read pair methods are the earliest methods developed for SV detection. BreakDancer [8], PEMer [36], VariationHunter [25, 26], MoDIL [39], MoGUL [40], and HYDRA [61] are based on this approach.

Split read approach : Unlike read depth and read pair approaches, only a few methods are developed based on the split read approach, such as Pindel [79], AGE [1] and Clipcrop [68].

Sequence assembly approach : A small amount of work has been done on this approach. The only methods based on sequence assembly are SOAPsv [46], Cortex [30] and NovelSeq [21].

Combining approach : There are a few methods that combine two or more approaches in one framework. This would usually increase the robustness and decrease the

false positive rate. CNVer [52], Genome STRiP [22], GASVPro [66], inGAP-SV [60], and CREST [71] are examples of such methods.

Chapter 3

SAGEcnv

In this chapter, we propose a new algorithm based on a read depth approach called *SAGEcnv*. This method needs to predict the number of times each short read appears in the donor genome, which is known as *copy count*. We use a modified implementation of a de novo assembly tool called SAGE to perform copy count estimation.

3.1 SAGE

SAGE [20] is a de novo assembler which uses an *overlap graph* to reconstruct the donor genome from short reads obtained from NGS platforms. Overlap graph is a graph platform which represents the overlapping of short reads. More precisely, in the overlap graph each read is denoted by a node and the overlapping of reads are denoted by edges between those reads. Since both of our methods are developed based on SAGE, in this section we present a brief explanation of this tool.

3.1.1 Building the overlap graph

Short reads generated by NGS platforms usually contain some errors (different platforms generate different types of errors such as substitutions or indels). Correcting these errors can help de novo assembler tools to reconstruct better contigs. This is usually done in a preprocessing step. SAGE uses a powerful error correction tool called RACER [29] to

```
Read r1  : AGCTAAGCATTTACGATAGCCGATAGCTAAATTAC
Read r2  : CGTAATTTAGCTATCGGCTATCGTAAATGCTTAGC
Read r3  : AACGTAATTTAGCTATCGGCTATCGTAAATGCTTA
Read r4  : GCATTTACGATAGCCGATAGCTAAATTACGTTATA
Read r5  : GTATAACGTAATTTAGCTATCGGCTATCGTAAATG
Read r6  : ATTTACGATAGCCGATAGCTAAATTACGTTATACT
Read r7  : TTTACGATAGCCGATAGCTAAATTACGTTATACTC
Read r8  : ATATAACGTAATTTAGCTATCGGCTATCGTAAATG
Read r9  : ATTTACGATAGCCGATAGCTAAATTACGTTATATA
Read r10 : CTATATAACGTAATTTAGCTATCGGCTATCGTAAA
```

Figure 3.1: A small example of a short read dataset [20].

fix errors in the short read datasets.

To build the overlap graph SAGE finds short reads that overlap with each other. Two strings overlap with each other if a prefix of one of them is identical to a suffix of the other. SAGE looks for maximum overlaps between two reads. Due to double stranded nature of DNA, one read can overlap with the reverse complement of the other read. In the overlap graph, the nodes represent short reads. If there is an overlap between two reads r_1 and r_2, an edge will be inserted to the overlap graph that connects r_1 and r_2. Each edge in the overlap graph is bi-directed, which means it has two arrowheads; one at each endpoint. This is designed to capture the double stranded structure of DNA. The type of this edge depends on the type of the overlap between the reads.

To see a small example, consider the short reads in Figure 3.1. The overlaps between these short reads are displayed in Figure 3.2 and the corresponding overlap graph is shown in Figure 3.3. A naive way of finding all overlaps between every pair of reads in a dataset of n reads has time complexity of $O(n^2)$. However, SAGE reduces this complexity by using a hash table to efficiently find all overlaps.

After inserting all edges which correspond to overlaps between reads, some of the edges are redundant because the information stored in them is already presented in the graph. In Figure 3.3 such edges are drawn with dashed lines which are called *transitive edges*. A linear time algorithm was proposed in [57] to remove transitive edges from an overlap graph. However, this algorithm can be utilized only when the whole overlap graph is created. SAGE uses a modified version of the algorithm that removes transitive edges

```
Read r1    :  AGCTAAGCATTTACGATAGCCGATAGCTAAATTAC
Read r̄2    :   GCTAAGCATTTACGATAGCCGATAGCTAAATTACG
Read r̄3    :     TAAGCATTTACGATAGCCGATAGCTAAATTACGTT
Read r4    :        GCATTTACGATAGCCGATAGCTAAATTACGTTATA
Read r̄5    :         CATTTACGATAGCCGATAGCTAAATTACGTTATAC
Read r6    :          ATTTACGATAGCCGATAGCTAAATTACGTTATACT
Read r7    :           TTTACGATAGCCGATAGCTAAATTACGTTATACTC
Read r̄8    :         CATTTACGATAGCCGATAGCTAAATTACGTTATAT
Read r9    :          ATTTACGATAGCCGATAGCTAAATTACGTTATATA
Read r̄10   :           TTTACGATAGCCGATAGCTAAATTACGTTATATAG
```

Figure 3.2: Overlapping of short reads of Figure 3.1. A read can overlap with another read or its reverse complement [20].

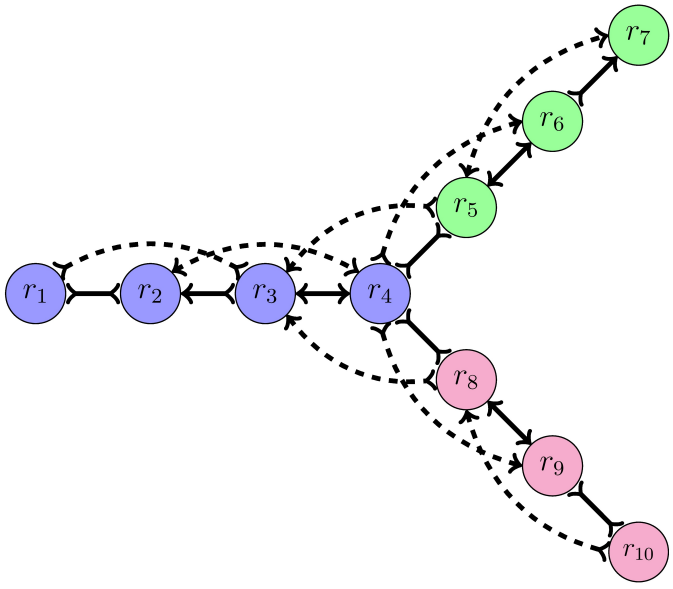

Figure 3.3: The overlap graph constructed based on the overlaps in Figure 3.2. Dashed-lines show the transitive edges [20].

during overlap graph construction. This dramatically reduces the memory required and makes SAGE capable of assembling large genomes. Therefore, the overlap graph does not have any transitive edges at the end of the overlap graph construction (see Figure 3.4).

Since the graph does not contain transitive edges, most of the nodes in the constructed overlap graph have degree 2. In Figure 3.4, r_2 and r_3 are examples of such nodes. These nodes can be removed because there is only one path containing them. For instance,

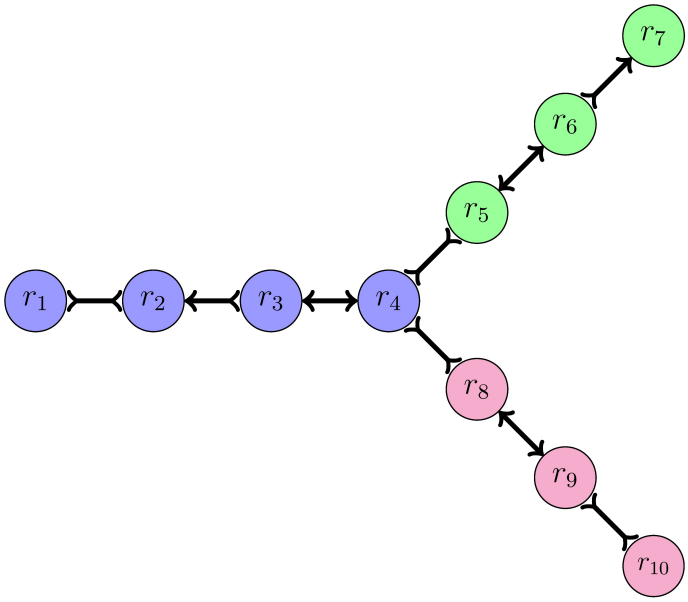

Figure 3.4: The overlap graph constructed by SAGE which does not contain any transitive edges [20].

nodes r_2 and r_3 can be removed because there is only one path between r_1 and r_4. In other words, the path between r_1 and r_4 can be contracted by removing nodes r_2 and r_3. SAGE performs this so-called *contracting composite paths* to reduce the overlap graph to a simpler graph. In this graph, each edge is actually a contig that is obtained by SAGE. Figure 3.5 shows the reduced graph which corresponds to Figure 3.4 obtained by contracting its composite paths.

3.1.2 Using Minimum Cost Flow to compute copy counts

To reconstruct the genome as closely as possible to the donor genome, the number of times that each contig appears in the genome should be calculated. In order to do this, an algorithm was proposed in [51] which converts this problem to an instance of the *Minimum Cost Flow (MCF)* problem. In the MCF problem, given a graph G, a flow upper bound $UB(e)$ for each edge e, and a cost $c(e)$ for each edge e, the goal is to assign a flow $f(e)$ to each edge e that minimizes the **total cost**, i.e. :

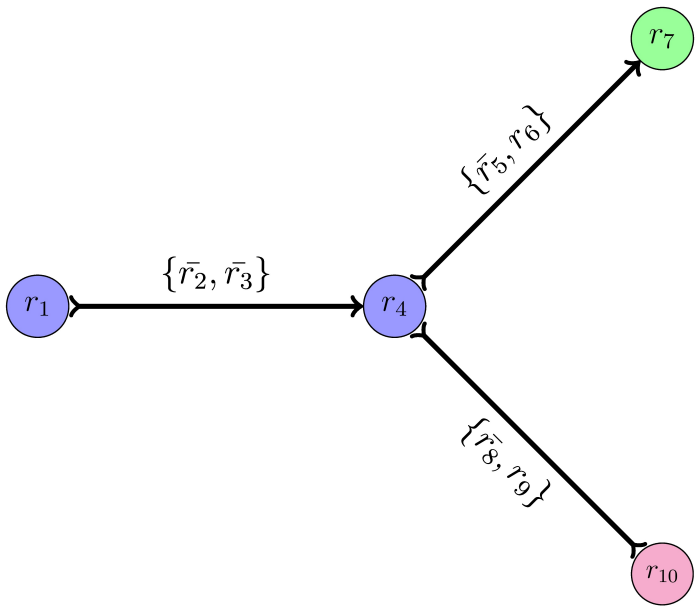

Figure 3.5: The overlap graph after contracting composite paths. Each edge stores some information about the reads that are represented by that edge [20].

$$\text{minimize} \sum_{\forall e} c(e)f(e). \tag{3.1}$$

The proposed algorithm in [51] uses a cost function for each edge in the overlap graph which maximizes the likelihood that the corresponding contig appears as many times as it appears in the donor genome. SAGE improves this idea by using a statistical analysis to set flow lower bounds for the edges. Then an MCF solver called CS2 [17] is used to find a solution to the MCF instance. This solution shows how many times each contig should appear in the assembled genome.

3.1.3 Final contigs extraction

The contigs in the overlap graph can be joined together to make larger sequences called scaffolds. Paired-end (mate pair) information is used for this purpose. If there is a paired-end read with one end in one contig and the other end in another contig, that paired-end is *supporting* joining those two contigs. If there are several paired-end reads that support joining the same contigs, those contigs are joined together. Sometimes there

is a gap between two contigs which will be filled with the ambiguous character 'N'. At the end, scaffolds are reported in FASTA format [1].

3.2 Copy count estimation

Our algorithm for detecting CNV needs to estimate copy counts of short reads. In order to obtain copy counts of the reads, the flow in the overlap graph, which is calculated by the Minimum Cost Flow solver, can be used. This flow gives an estimated copy number of each contig, because the cost function of the MCF instance was defined in such a way that maximizes the likelihood that the number of flow units on this contig is the same as the number of times that it appears in the donor genome.

Therefore, the copy count of each read r can be extracted by finding the contig that contains r and assigning the flow of that contig as its copy count. An important point here is that SAGE uses unique reads, meaning it considers each read only once. As a result, we also work with unique reads. Sometimes a particular unique read can appear in more than one contig. In this case, the copy count of that read is defined as the sum of the flows of all contigs which contain that read.

3.3 Short read alignment

Due to the existence of repeats and segmental duplications in the human genome, a fraction of the short reads will not be aligned to the reference genome uniquely. Although by utilizing the paired-end information more reads can be mapped uniquely, still a small fraction of reads can map to multiple locations of the reference genome. This presents a challenge to the read depth approach, as this approach works mainly by aligning short reads to the reference genome.

Different methods have dealt with this challenge differently. A few tools have ignored such short reads completely (which is clearly not a good choice). Some choose one of the possible alignment positions randomly. MAQ [43] and Bowtie [38] are instances of

[1] For information about FASTA format, please check http://en.wikipedia.org/wiki/FASTA_format

short read aligners which make such a random choice. This sort of handling of multiply aligned short reads is much better than ignoring those reads and does not effect the read depth. However, they still ignore all mappings of each read.

There are a few methods that deal with all "good" mappings of each short read to improve the sensitivity within duplicated regions. These alignments can be obtained using some read alignment tools such as Bowtie, BWA [42], mrFAST [4], and mrsFAST [19]. We use Bowtie to align all unique short reads to the reference genome to find all possible alignments.

3.4 CNV calling

Having estimated the copy counts as well as all possible alignments of each **unique** short read we can use this information for CNV detection. By unique short read, we mean that each short read is processed only once, even if it appears many times in the read dataset. The basic idea is that for normal regions (non CNV regions) with possibly a few repeats in the reference genome, it is expected that the estimated copy count of unique reads sequenced from that region is almost equal to the number of alignments of those unique reads. Otherwise, if there is a deletion (insertion) in the donor genome, the estimated copy count is expected to be less (more) than the number of alignments.

As a result, for each read we can define a ratio that can be considered as the depth of that read in the reference genome. Denoting, for each read r, the estimated copy count by r_{cc} and the number of times it aligns to the reference genome by r_{na}, the depth of coverage of read r is defined as :

$$r_{doc} = \frac{r_{cc}}{r_{na}}. \tag{3.2}$$

If r_{doc} is equal to 1, it means the read r is sequenced from a normal region. If r_{doc} is greater than 1, it means the read r is sequenced from a region with more copies in the donor genome than the reference genome (insertion) and if r_{doc} is less than 1, it means the read r is sequenced from a region with fewer copies in the donor genome than the reference genome (deletion).

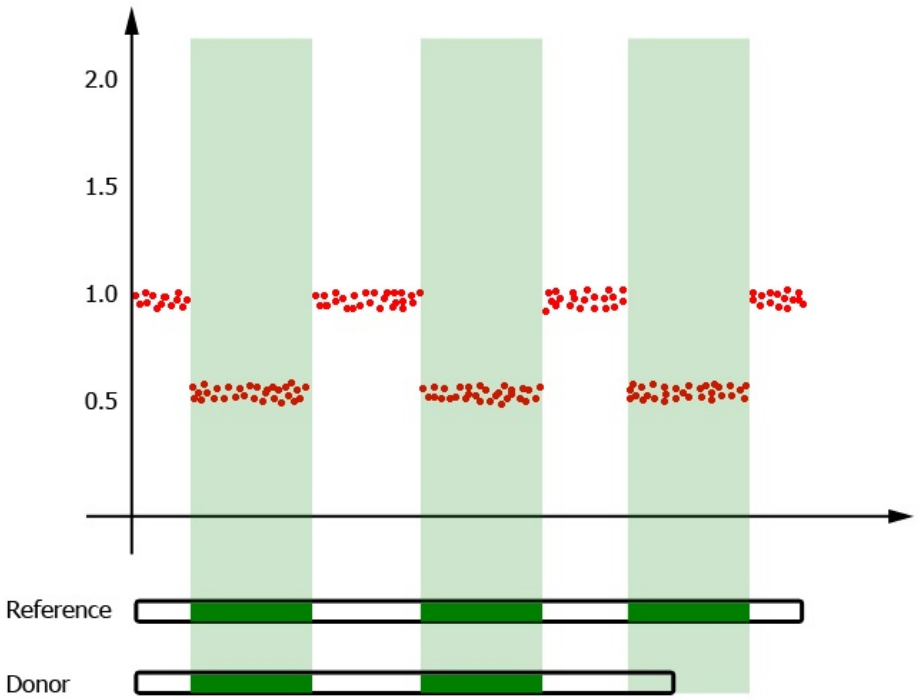

Figure 3.6: The distribution of depth of coverage (DOC) in the reference genome in the case of deletion (loss in copy number). In this figure, vertical axis is the DOC and therefore, each point shows the DOC of a window in the reference genome.

We consider windows of fixed size in the reference genome and calculate the sum of r_{doc} values of all reads in each window. This number can be considered as the depth of coverage (DOC) of that window. As mentioned in the Chapter 2, the coverage of NSG sequencing platforms is biased by GC-content. Therefore, the depth of coverage of windows should be adjusted to remove the effect of GC-content. We perform this correction according to the following formula :

$$DOC'_i = DOC_i \times \frac{m}{m_{GC}} , \tag{3.3}$$

where DOC_i is the depth of coverage of the i-th window, m_{GC} is the median depth of coverage of all windows that have the same GC-content as the i-th window, and m is the overall median depth of coverage of all the windows.

Based on the definition of r_{doc}, we can deduce that the regions that have a loss of copy number should have a drop in the corresponding windows DOC. Figure 3.6 illustrates this

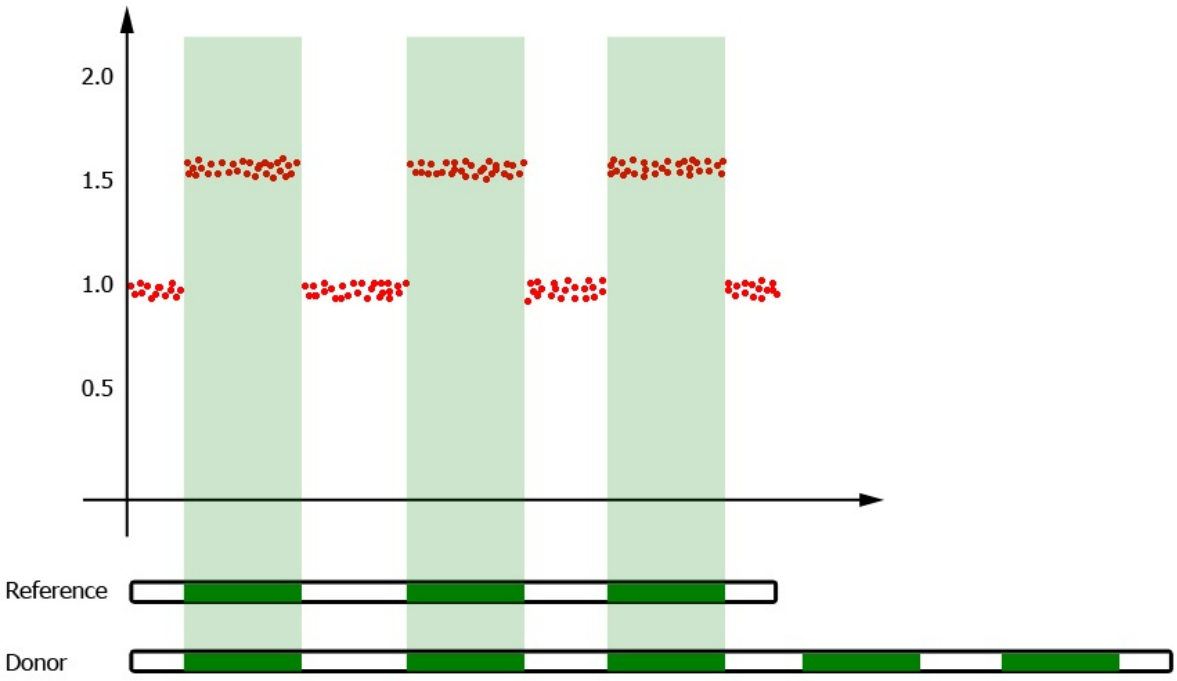

Figure 3.7: The distribution of DOC in the reference genome in the case of insertion (gain in copy number). In this figure, vertical axis is the DOC and therefore, each point shows the DOC of a window in the reference genome.

case. Conversely, regions that have a gain of copy number have a rise in the corresponding windows DOC which is shown in Figure 3.7.

Using the GC-adjusted depth of coverage of windows, a change point analysis can be done to find regions that have significantly higher or significantly lower depth of coverage than the average as the detected CNV. In order to perform CNV calling, we use the tool proposed in RDxplorer [80] which uses a significance test to detect CNVs. The general steps of SAGEcnv are illustrated as pseudo-code in Algorithm 1. Later in the Chapter 5, the accuracy of SAGEcnv algorithm will be evaluated by comparing with other methods.

Algorithm 1: Pseudo-code for SAGEcnv.

input : • short reads obtained from a donor genome using an NGS platform
• the reference genome

output: copy number variations, their type and breakpoints

estimate copy count of each unique read
for *each unique read* r **do**
    `/* align read` r `to the reference genome */`
    $positions[1..m] \leftarrow$ all mapping positions of read r
    $count \leftarrow$ estimated copy count of read r
    `/* depth of coverage calculation */`
    for $i \leftarrow 1$ **to** m **do** `// process all alignments`
        $depth[positions[i]] \leftarrow depth[positions[i]] + \frac{count}{m}$
    end
    `/* windows depth of coverage calculation */`
    $len \leftarrow$ length of the genome
    $winSize \leftarrow$ the desired size of windows
    $winNo \leftarrow \frac{len}{winSize}$
    for $i \leftarrow 1$ **to** $winNo$ **do** `// process every window`
        for $j \leftarrow 1$ **to** $winSize$ **do**
            $winDepth[i] += depth[i * 100 + j]$
        end
    end
    `/* Call EWT method [80] to detect CNVs */`
    `EWT(`$winDepth$`)`
end

Chapter 4

SAGEsv

As mentioned in the discussion about the existing work on the SV detection problem in Section 2.3, little has been done using the sequence assembly approach. This is because de novo assembly is a challenging problem (due to the existence of repeats and segmental duplications) and hence, structural variation detection using the other approaches seems a much easier task.

However, the sequence assembly approach has some advantages over other approaches. There are some types of variations that cannot be found by other approaches among which novel insertion is the most important one. Methods based on sequence assembly are able to find novel insertions as well as any other types of variations. Moreover, when sequence assembly based methods become mature enough, they will be the best tools to find variations in the donor genome because not only they can determine the types of variations, but also they provide information about the content and the exact breakpoints of all of them.

These facts provide a high level of motivation to work more on the sequence assembly approach. In this chapter, we introduce a new method called *SAGEsv*, also based on the SAGE assembler. A brief explanation of SAGE was presented in Section 3.1. Here we just describe how it is modified to find structural variations.

4.1 Utilizing the reference genome

The basic idea of SAGEsv is to use more information from the reference genome. As it was mentioned earlier, de novo assemblers do not use, by definition, any information from the reference genome. Thus, it is expected that if we use some information from the reference genome, we can attain a better assembly.

This is due to the fact that the donor genome is supposed to be very similar to the reference genome. On the other hand, the reference genome should be only used as a guide and not for assembling the sequence. That is because the differences between the donor genome and the reference genome are exactly those regions that are interesting to study from the viewpoint of SV detection. Therefore, we use the reference genome just as a guide and still use the overlap graph (which is built based on the donor genome) to find sequences.

4.2 Anchors

SAGEsv uses the reference genome in order to fill the gaps between contigs and join them to get longer contigs. Suppose that all the steps of SAGE are completed, i.e., the overlap graph is build from short reads, the Minimum Cost Flow problem is solved, the corresponding flow of each edge is calculated, and the paired-end information is used to join contigs to get scaffolds. The result is an overlap graph in which every edge corresponds to one scaffold. We try to join some of these scaffolds to get even longer ones.

Each scaffold is aligned to the reference genome using BWA and all mapping positions of that scaffold are stored. It is expected that many of these scaffolds appear only once in the reference genome. A scaffold is called *unique* if it appears only once in the reference genome and the flow of its corresponding edge is 1. In other words, a unique scaffold appears only once in both the reference genome and the donor genome. We consider these unique scaffolds as *anchors* in the reference genome. In addition, we assume that pairs of consecutive unique scaffolds are usually aligned to the reference genome in the same

order in which they appear in the donor genome. This assumption can be wrong in some cases, when unique scaffolds appear in translocated or inverted regions. Nevertheless, whenever the assumption is true, if we can find a "good" path in the **donor genome** between two scaffolds, they can be joined to make a larger scaffold.

4.3 Finding paths

For a particular pair of consecutive unique scaffolds us_1 and us_2, the overlap graph is explored to find all paths between these unique scaffolds. However, since the overlap graph can be very big, some restrictions are needed to make it computationally possible. As we have the mapping positions of all edges, we can easily determine three sets of edges:

1. Those edges whose corresponding scaffolds are mapped to the reference genome in the region between us_1 and us_2. These edges are stored in set A.

2. The edges whose corresponding scaffolds are mapped to the reference genome but not to the region between us_1 and us_2. They are stored in set B.

3. The edges whose corresponding scaffolds do not map to the reference genome (these may be novel sequences in the donor genome). They are stored in set C.

When exploring the overlap graph, we are only interested in those paths that have a small overlap with sets B and C. In addition, the length of the path should be close to the distance between mappings of unique scaffolds in the reference genome. Supposing the distance between mapping positions of us_1 and us_2 is indicated by d, only those paths that have all of the three conditions below will be accepted as valid paths:

1. At most $0.1 \times d$ bp of its sequence belongs to the set B.

2. At most $0.1 \times d$ bp of its sequence belongs to the set C.

3. Its length is in the range of $[0.8 \times d, 1.2 \times d]$.

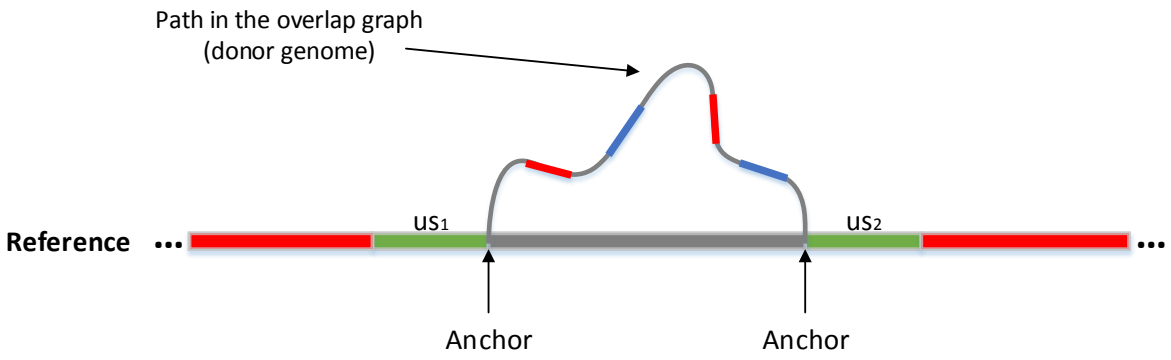

Figure 4.1: Finding a path in the overlap grpah between two consecutive unique scaffolds. In this figure, unique scaffolds have color green. The region between these scaffolds is gray and the regions which are not between unique scaffolds are shown in red. In addition, new sequences (novel insertions) are shown in blue. A valid path should use more from the gray region and just a small fraction of the red and blue regions.

Figure 4.1 shows an example of such a path. We do not pose any restriction on the fraction of the path belonging to the set A because we expect the sequence of the donor genome between the two scaffolds to be close to the sequence of the reference genome between the two scaffolds. By applying these restrictions to the path finding process, it becomes faster. We implemented a backtracking algorithm constrained by the above conditions, to find all the *simple paths* in the overlap graph (donor genome) between us_1 and us_2. A simple path is a path that contains each node at most once.

4.4 Merging scaffolds

If there is a unique path between two consecutive unique scaffolds, we will join those scaffolds by the unique path between them. In this case, the whole path will be converted to a new edge e_{new} by merging all the edges in that path starting form one unique scaffold and ending at the other one.

Every time an edge e is merged with e_{new}, we look at its flow. If the flow is 1, edge e will be deleted after merging. If the flow is greater than 1, edge e will not be deleted after merging but one unit will be deducted from its flow. In other words, the process of merging the path into the new edge e_{new} should be a *flow-obeying* process.

To process all the unique scaffolds, we sort them based on their mapping positions in the reference genome. Then, two consecutive unique scaffolds with the smallest mapping positions are selected. If there is a unique path between them, the scaffolds are joined together using that path and the next pair of consecutive scaffolds is processed.

4.5 Extracting variations

The final overlap graph contains improved scaffolds which give a better assembly of the donor genome. These scaffolds can be used to extract SVs by a careful comparison with the reference genome. We use a tool developed in [46] to detect structural variations. The general idea is to first align scaffolds to the reference genome using BLAT [32], to find out the source chromosome of each scaffold. Next, structural variations are detected using a more careful alignment. For SVs smaller than 50 bp BWA [42] is employed and for longer SVs (up to 50 Kb) LASTZ [23] is used.

Chapter 5

Comparison

In this chapter we compare our algorithms with previously proposed methods. Our CNV detection algorithm, SAGEcnv, which is based on the read depth approach is compared with RDxplorer [80] and CNVer [52]. SAGEsv is compared to SOAPsv [46] since they are both based on the sequence assembly approach.

5.1 A discussion about validation sets

An important point, which should be noted here, is that no true "gold standard" for structural variations has been reported yet [69]. It means there is no comprehensive validation set of SVs/CNVs that a) contains all of the real SVs of a particular genome and b) does not contain any incorrect SVs. More surprisingly, suggested gold standards which contain SVs validated by experimental methods are not necessarily consistent. For example, a study on two suggested gold standards, Kidd *et al.* [34] and Conrad *et al.* [12], reported relatively poor consistency between them [82].

On the other hand, different SV detection methods are complementary as it is also reported in the 1000 genomes project's paper [56]. In this project 19 different SV discovery methods were used. For each of these methods, there is a set of SV calls that can be detected only by that particular method (refer to the supplementary material of [56]).

The lack of true gold standards makes the evaluation of SV detection methods difficult. Nevertheless, previously published papers have usually used a set of SV calls obtained

by another method (or methods) for the same individual or a population of individuals containing that individual, which are validated by experimental methods. Although such sets contain SVs that are highly likely true SVs for the case study, it is quite possibly not a comprehensive set of SVs, i.e. there are many real SV calls that are not included in those sets.

5.2 Datasets and experiments

5.2.1 SAGEcnv

In order to evaluate SAGEcnv, we used a dataset of short reads sampled from the individual NA18507, which can be obtained from NCBI[1] under accession number SRA000271. This dataset contains about 3.5 billion short reads of length 36 bp with a mean insert size of 208 bp. The exact same dataset is also used in the experiments reported by RDxplorer [80] and CNVer [52], which allows us to directly compare our results with theirs. Instead of working on the whole genome, the Bowtie [38] aligner was used to find those short reads that can be aligned to chromosome 1 of the reference genome. These short reads were considered as the input dataset for our evaluation.

This dataset was used as input for SAGEcnv. We used Bowtie to find all possible mappings of each short read allowing at most 3 mismatches. Then, after estimating the copy count of each short read using a modified version of SAGE, SAGEcnv was run to find CNV calls (see Algorithm 1). We used a window size of 175, which was chosen carefully based on the performed experiments. The results are presented in the next section.

5.2.2 SAGEsv

To assess the accuracy of SAGEsv, an experiment on the simulated data obtained from chromosome 17 of the HuRef assembly [41] was performed. We decided to work on synthetic data because if we use real dataset of short reads, it is not possible to specify

[1] http://www.ncbi.nlm.nih.gov

the origin of each short read (i.e. we do not know which chromosome each short reads comes from). In other words, in order to use real dataset, the whole dataset should be processed which needs a lot of effort. Therefore, we decided to evaluate the performance of our method on the synthetic data at this point. Moreover, the advantage of using the HuRef dataset is that a set of variations has been available which is validated by experimental methods. Although this gold standard does not contain all of the SVs, we still can use it to evaluate our SV discovery method.

All of the data related to HuRef can be obtained freely from the Craig Venter Institute website[2]. We performed our experiments on chromosome 17. An Illumina simulator called *pIRS* [28] was used to generate a dataset of synthetic short reads of length 100 bp and mean insert size of 300. We used this dataset to run both SAGEsv and SOAPsv for comparison. The results are presented in the next section.

5.3 Results

5.3.1 SAGEcnv

The results of CNV detection from chromosome 1 of individual NA18507 using RDxplorer, CNVer and SAGEcnv are shown in Table 5.1. In this table, the number of calls shows the number of detected CNVs using each method, each of which can be a true CNV region or a false one. In order to evaluate these methods, detected CNVs are compared with two validation sets. The first validation set is Genome Structural Variation Consortium database (GSV)[3], which is a database of 8599 CNV regions found in 40 individuals using experimental methods. The second one is the gold standard obtained previously for the same individual NA18507 by Kidd *et al.* [34].

For GSV calls, we have reported the percentage of predicted CNV calls validated by GSV (those which have an overlap with one of the calls in GSV) as well as the percentage of bases in CNV calls validated by GSV. For Kidd *et al.* calls, we have reported the percentage of CNV calls in the gold standard that are detected by each method.

[2]http://huref.jcvi.org
[3]http://projects.tcag.ca/variation/ng42m_cnv.php

Table 5.1: Comparison of CNV calls detected by three methods for chromosome 1 of NA18507. We show the number of detected calls, the percentage of chromosome 1 bases covered by CNV calls, the percentage of CNV calls validated by GSV, the percentage of CNV bases validated by GSV, and the percentage of Kidd *et al.* calls detected by these methods.

Method	Number of calls	Coverage (%)	Against GSV (by calls)	Against GSV (by bases)	Against Kidds calls
SAGEcnv	427	2	45.9	63.2	90.9
Rdxplorer	1151	2.45	34	54	82
CNVer	435	1.75	73	62	82

Although these results show that SAGEcnv detects fewer calls from GSV, more bases of the detected calls are validated by GSV calls. This means CNV calls detected by SAGEcnv have more overlaps with GSV calls. In addition, SAGEcnv performs better in detecting the Kidd *et al.* calls. RDxplorer predicts more CNV calls but has a high False Detection Rate (FDR). It should also be noted that CNVer uses both a read depth and a read pair approaches for CNV discovery, which helps it to attain better sensitivity.

5.3.2 SAGEsv

To compare SAGEsv with SOAPsv, we analyzed deletions and insertions separately. In addition, since these programs are based on sequence assembly approach and are able to find variations of any size, to perform a better analysis we consider different size ranges for comparing. Table 5.2 shows the comparison for deletions. Since we are using the variations reported by HuRef's website as the gold standard, the analysis is easier than in the case of SAGEcnv.

This table reports the percentage of gold standard SV calls (in each size range) that are detected by each method. There are two commonly used conventions for considering a deletion as detected. The first one is to consider a deletion as detected if one of the predicted deletions overlaps with it by at least 1 bp. The second one is to consider a deletion as detected if at least 50% of its bases overlap with the predicted deletions. The results show better performance of SAGEsv in both measures, except for the deletion in

Table 5.2: Comparison of SOAPsv and SAGEsv in detecting deletions with different sizes in the synthetic dataset obtained from chromosome 17 of the HuRef genome. In each case, the better result is shaded.

		SOAPsv		SAGEsv	
size range	number of true SVs	overlap (%)	%50 overlap (%)	overlap (%)	%50 overlap (%)
1, 20	8402	14.9	5.5	14.8	5.6
20, 50	226	27.9	16.8	30.5	18.1
50, 100	63	25.4	22.2	34.9	27
100, 1000	128	37.5	35.9	39.1	35.9
1000, max	9	11.1	11.1	22.2	22.2

Table 5.3: Comparison of SOAPsv and SAGEsv in detecting insertions with different sizes in the synthetic dataset obtained from chromosome 17 of the HuRef genome. In each case, the better result is shaded.

		SOAPsv	SAGEsv
size range	number of true SVs	close breakpoints (%)	close breakpoints (%)
1, 20	8183	71.3	78.1
20, 50	256	14.4	15.6
50, 100	59	15.3	37.3
100, 1000	91	24.2	42.9
1000, max	16	18.75	50

the size range $[1, 20]$.

The comparison of the methods for insertions is presented in Table 5.3. As in the case of deletions, this table shows the percentage of SV calls detected by the methods. However, since insertions are known by only a single breakpoint in the reference genome, the definition of detection should be different. An insertion in the gold standard is considered as detected if at least one of the predicted insertions is "close enough" to it. Here we consider two insertions to be close enough if the distance between them is at most 300 bp. The same definition of close breakpoints is used in another study of structural variation detection [27].

The comparison shows that SAGEsv is better than SOAPsv in all size ranges. More interestingly, the difference is bigger for longer insertions. This clearly demonstrates the higher power of SAGEsv for detection of long insertions.

Chapter 6

Conclusions

Since several studies have shown association of structural variations to some diseases, extensive research has been conducted on SV discovery during the past few years. In this thesis we proposed new ideas for SV/CNV detection based on the SAGE de novo assembler. We showed how the proposed method for CNV detection, SAGEcnv, uses estimated copy counts to detect CNVs more accurately than an existent method based on read depth, RDxplorer. To the best of our knowledge, this is the first use of estimated copy counts of short reads to solve a computational biologic problem.

In the other proposed method, SAGEsv, we tried to improve the assembled sequences produced by SAGE using the reference genome as a guide, and then used them for SV detection. We compared our method with SOAPsv, which is also based on sequence assembly and showed the better performance of SAGEsv. The advantage of sequence assembly approaches is that it can detect any types of variations and give any information such as size, content and location of each variation. However, more work needs to be done on such methods since they are not mature enough at this point.

We believe that the proposed algorithms could be improved. Possible future work for SAGEcnv is using other signatures like read pair to improve sensitivity. Usually methods which combine different approaches provide better accuracy. On the other hand, for SAGEsv there are several ideas worth trying. Maybe the most important one is designing a new method for extracting variations from sequenced assemblies, because the proposed method used in SOAPsv is very slow and difficult to use on large genomes

such as the human genome. Another possible idea is to try to find more complex paths between unique scaffolds instead of simple paths. In addition, in the case when more than one path are found, it may be still possible to choose one of them to fill the gap between unique scaffolds and merge them.

Bibliography

[1] Alexej Abyzov and Mark Gerstein. AGE: defining breakpoints of genomic structural variants at single-nucleotide resolution, through optimal alignments with gap excision. *Bioinformatics*, 27(5):595–603, 2011.

[2] Alexej Abyzov, Alexander E Urban, Michael Snyder, and Mark Gerstein. CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome research*, 21(6):974–984, 2011.

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

[4] Can Alkan, Jeffrey M Kidd, Tomas Marques-Bonet, Gozde Aksay, Francesca Antonacci, Fereydoun Hormozdiari, Jacob O Kitzman, Carl Baker, Maika Malig, Onur Mutlu, et al. Personalized copy number and segmental duplication maps using next-generation sequencing. *Nature genetics*, 41(10):1061–1067, 2009.

[5] David R Bentley, Shankar Balasubramanian, Harold P Swerdlow, Geoffrey P Smith, John Milton, Clive G Brown, Kevin P Hall, Dirk J Evers, Colin L Barnes, Helen R Bignell, et al. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218):53–59, 2008.

[6] Peter J Campbell, Philip J Stephens, Erin D Pleasance, Sarah O'Meara, Heng Li, Thomas Santarius, Lucy A Stebbings, Catherine Leroy, Sarah Edkins, Claire Hardy, et al. Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing. *Nature genetics*, 40(6):722–729, 2008.

[7] Nigel P Carter. Methods and strategies for analyzing copy number variation using DNA microarrays. *Nature genetics*, 39:S16–S21, 2007.

[8] Ken Chen, John W Wallis, Michael D McLellan, David E Larson, Joelle M Kalicki, Craig S Pohl, Sean D McGrath, Michael C Wendl, Qunyuan Zhang, Devin P Locke, et al. BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nature methods*, 6(9):677–681, 2009.

[9] Derek Y Chiang, Gad Getz, David B Jaffe, Michael JT O'Kelly, Xiaojun Zhao, Scott L Carter, Carsten Russ, Chad Nusbaum, Matthew Meyerson, and Eric S Lander. High-resolution mapping of copy-number alterations with massively parallel sequencing. *Nature methods*, 6(1):99–103, 2008.

[10] Suzanne Clancy. Copy number variation. *Nature Education*, 1(1), 2008.

[11] FS Collins, ES Lander, J Rogers, RH Waterston, and IHGS Conso. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, 2004.

[12] Donald F Conrad, Dalila Pinto, Richard Redon, Lars Feuk, Omer Gokcumen, Yujun Zhang, Jan Aerts, T Daniel Andrews, Chris Barnes, Peter Campbell, et al. Origins and functional impact of copy number variation in the human genome. *Nature*, 464(7289):704–712, 2009.

[13] The 1000 Genomes Project Consortium. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073, 2010.

[14] Evan E Eichler and Andrew W Zimmerman. A hot spot of genetic instability in autism. *New England Journal of Medicine*, 358(7):737, 2008.

[15] Kelly A Frazer, Dennis G Ballinger, David R Cox, David A Hinds, Laura L Stuve, Richard A Gibbs, John W Belmont, Andrew Boudreau, Paul Hardenbol, Suzanne M Leal, et al. A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449(7164):851–861, 2007.

[16] Santhosh Girirajan, Catarina D Campbell, and Evan E Eichler. Human copy number variation and complex genetic disease. *Annual review of genetics*, 45:203–226, 2011.

[17] Andrew V Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of algorithms*, 22(1):1–29, 1997.

[18] Enrique Gonzalez, Hemant Kulkarni, Hector Bolivar, Andrea Mangano, Racquel Sanchez, Gabriel Catano, Robert J Nibbs, Barry I Freedman, Marlon P Quinones, Michael J Bamshad, et al. The influence of CCL3L1 gene-containing segmental duplications on HIV-1/AIDS susceptibility. *Science*, 307(5714):1434–1440, 2005.

[19] Faraz Hach, Fereydoun Hormozdiari, Can Alkan, Farhad Hormozdiari, Inanc Birol, Evan E Eichler, and S Cenk Sahinalp. mrsfast: a cache-oblivious algorithm for short-read mapping. *Nature methods*, 7(8):576–577, 2010.

[20] Md. Bahlul Haider. *A new algorithm for de novo genome assembly.* PhD thesis, The University of Western Ontario, 2012.

[21] Iman Hajirasouliha, Fereydoun Hormozdiari, Can Alkan, Jeffrey M Kidd, Inanc Birol, Evan E Eichler, and S Cenk Sahinalp. Detection and characterization of novel sequence insertions using paired-end next-generation sequencing. *Bioinformatics*, 26(10):1277–1283, 2010.

[22] Robert E Handsaker, Joshua M Korn, James Nemesh, and Steven A McCarroll. Discovery and genotyping of genome structural polymorphism by sequencing on a population scale. *Nature genetics*, 43(3):269–276, 2011.

[23] Robert S Harris. *Improved pairwise alignment of genomic DNA.* PhD thesis, The Pennsylvania State University, 2007.

[24] Ingo Helbig, Heather C Mefford, Andrew J Sharp, Michel Guipponi, Marco Fichera, Andre Franke, Hiltrud Muhle, Carolien de Kovel, Carl Baker, Sarah von Spiczak, et al. 15q13. 3 microdeletions increase risk of idiopathic generalized epilepsy. *Nature genetics*, 41(2):160–162, 2009.

[25] Fereydoun Hormozdiari, Can Alkan, Evan E Eichler, and S Cenk Sahinalp. Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome research*, 19(7):1270–1278, 2009.

[26] Fereydoun Hormozdiari, Iman Hajirasouliha, Phuong Dao, Faraz Hach, Deniz Yorukoglu, Can Alkan, Evan E Eichler, and S Cenk Sahinalp. Next-generation VariationHunter: combinatorial algorithms for transposon insertion discovery. *Bioinformatics*, 26(12):i350–i357, 2010.

[27] Fereydoun Hormozdiari, Iman Hajirasouliha, Andrew McPherson, Evan E Eichler, and S Cenk Sahinalp. Simultaneous structural variation discovery among multiple paired-end sequenced genomes. *Genome research*, 21(12):2203–2212, 2011.

[28] Xuesong Hu, Jianying Yuan, Yujian Shi, Jianliang Lu, Binghang Liu, Zhenyu Li, Yanxiang Chen, Desheng Mu, Hao Zhang, Nan Li, et al. pirs: profile-based illumina pair-end reads simulator. *Bioinformatics*, 28(11):1533–1535, 2012.

[29] Lucian Ilie and Michael Molnar. Racer: Rapid and accurate correction of errors in reads. *Bioinformatics*, 29(19):2490–2493, 2013.

[30] Zamin Iqbal, Mario Caccamo, Isaac Turner, Paul Flicek, and Gil McVean. De novo assembly and genotyping of variants using colored de bruijn graphs. *Nature genetics*, 44(2):226–232, 2012.

[31] Sergii Ivakhno, Tom Royce, Anthony J Cox, Dirk J Evers, R Keira Cheetham, and Simon Tavaré. CNAseg–a novel framework for identification of copy number changes in cancer from second-generation sequencing data. *Bioinformatics*, 26(24):3051–3058, 2010.

[32] W James Kent. BLAT–the BLAST-like alignment tool. *Genome research*, 12(4):656–664, 2002.

[33] W James Kent, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler, and David Haussler. The human genome browser at UCSC. *Genome research*, 12(6):996–1006, 2002.

[34] Jeffrey M Kidd, Gregory M Cooper, William F Donahue, Hillary S Hayden, Nick Sampas, Tina Graves, Nancy Hansen, Brian Teague, Can Alkan, Francesca Antonacci, et al. Mapping and sequencing of structural variation from eight human genomes. *Nature*, 453(7191):56–64, 2008.

[35] Tae-Min Kim, Lovelace Luquette, Ruibin Xi, and Peter Park. rSW-seq: algorithm for detection of copy number alterations in deep sequencing data. *BMC bioinformatics*, 11(1):432, 2010.

[36] Jan O Korbel, Alexej Abyzov, Xinmeng Jasmine Mu, Nicholas Carriero, Philip Cayting, Zhengdong Zhang, Michael Snyder, Mark B Gerstein, et al. PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol*, 10(2):R23, 2009.

[37] Eric S Lander, Lauren M Linton, Bruce Birren, Chad Nusbaum, Michael C Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michael Doyle, William FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[38] Ben Langmead, Cole Trapnell, Mihai Pop, Steven L Salzberg, et al. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biol*, 10(3):R25, 2009.

[39] Seunghak Lee, Fereydoun Hormozdiari, Can Alkan, and Michael Brudno. MoDIL: detecting small indels from clone-end sequencing with mixtures of distributions. *Nature Methods*, 6(7):473–474, 2009.

[40] Seunghak Lee, Eric Xing, and Michael Brudno. MoGUL: detecting common insertions and deletions in a population. In *Research in Computational Molecular Biology*, pages 357–368. Springer, 2010.

[41] Samuel Levy, Granger Sutton, Pauline C Ng, Lars Feuk, Aaron L Halpern, Brian P Walenz, Nelson Axelrod, Jiaqi Huang, Ewen F Kirkness, Gennady Denisov, et al.

The diploid genome sequence of an individual human. *PLoS biology*, 5(10):e254, 2007.

[42] Heng Li and Richard Durbin. Fast and accurate long-read alignment with Burrows–Wheeler transform. *Bioinformatics*, 26(5):589–595, 2010.

[43] Heng Li, Jue Ruan, and Richard Durbin. Mapping short dna sequencing reads and calling variants using mapping quality scores. *Genome research*, 18(11):1851–1858, 2008.

[44] Ruiqiang Li, Hongmei Zhu, Jue Ruan, Wubin Qian, Xiaodong Fang, Zhongbin Shi, Yingrui Li, Shengting Li, Gao Shan, Karsten Kristiansen, et al. De novo assembly of human genomes with massively parallel short read sequencing. *Genome research*, 20(2):265–272, 2010.

[45] Wenli Li and Michael Olivier. Current analysis platforms and methods for detecting copy number variation. *Physiological genomics*, 45(1):1–16, 2013.

[46] Yingrui Li, Hancheng Zheng, Ruibang Luo, Honglong Wu, Hongmei Zhu, Ruiqiang Li, Hongzhi Cao, Boxin Wu, Shujia Huang, Haojing Shao, et al. Structural variation in two human genomes mapped at single-nucleotide resolution by whole genome de novo assembly. *Nature biotechnology*, 29(8):723–730, 2011.

[47] Lin Liu, Yinhu Li, Siliang Li, Ni Hu, Yimin He, Ray Pong, Danni Lin, Lihua Lu, and Maggie Law. Comparison of next-generation sequencing systems. *Journal of Biomedicine and Biotechnology*, 2012, 2012.

[48] Alberto Magi, Matteo Benelli, Seungtai Yoon, Franco Roviello, and Francesca Torricelli. Detecting common copy number variants in high-throughput sequencing data by using JointSLM algorithm. *Nucleic acids research*, 39(10):e65–e65, 2011.

[49] Elaine R Mardis. A decade's perspective on DNA sequencing technology. *Nature*, 470(7333):198–203, 2011.

[50] Christian R Marshall, Abdul Noor, John B Vincent, Anath C Lionel, Lars Feuk, Jennifer Skaug, Mary Shago, Rainald Moessner, Dalila Pinto, Yan Ren, et al. Structural variation of chromosomes in autism spectrum disorder. *The American Journal of Human Genetics*, 82(2):477–488, 2008.

[51] Paul Medvedev and Michael Brudno. Ab initio whole genome shotgun assembly with mated short reads. In *Research in Computational Molecular Biology*, pages 50–64. Springer, 2008.

[52] Paul Medvedev, Marc Fiume, Misko Dzamba, Tim Smith, and Michael Brudno. Detecting copy number variation with mated short reads. *Genome research*, 20(11):1613–1622, 2010.

[53] Paul Medvedev, Monica Stanciu, and Michael Brudno. Computational methods for discovering structural variation with next-generation sequencing. *Nature methods*, 6:S13–S20, 2009.

[54] Heather C Mefford, Séverine Clauin, Andrew J Sharp, Rikke S Moller, Reinhard Ullmann, Raj Kapur, Dan Pinkel, Gregory M Cooper, Mario Ventura, H Ropers, et al. Recurrent reciprocal genomic rearrangements of 17q12 are associated with renal disease, diabetes, and epilepsy. *The American Journal of Human Genetics*, 81(5):1057–1069, 2007.

[55] Michael L Metzker. Sequencing technologiesthe next generation. *Nature Reviews Genetics*, 11(1):31–46, 2009.

[56] Ryan E Mills, Klaudia Walter, Chip Stewart, Robert E Handsaker, Ken Chen, Can Alkan, Alexej Abyzov, Seungtai Chris Yoon, Kai Ye, R Keira Cheetham, et al. Mapping copy number variation by population-scale genome sequencing. *Nature*, 470(7332):59–65, 2011.

[57] Eugene W Myers. The fragment assembly string graph. *Bioinformatics*, 21(suppl 2):ii79–ii85, 2005.

[58] Zemin Ning, Anthony J Cox, and James C Mullikin. SSAHA: a fast search method for large DNA databases. *Genome research*, 11(10):1725–1729, 2001.

[59] Nathan Pankratz, Alexandra Dumitriu, Kurt N Hetrick, Mei Sun, Jeanne C Latourelle, Jemma B Wilk, Cheryl Halter, Kimberly F Doheny, James F Gusella, William C Nichols, et al. Copy number variation in familial Parkinson disease. *PloS one*, 6(8):e20988, 2011.

[60] Ji Qi and Fangqing Zhao. inGAP-sv: a novel scheme to identify and visualize structural variation from paired end mapping data. *Nucleic acids research*, 39(suppl 2):W567–W575, 2011.

[61] Aaron R Quinlan, Royden A Clark, Svetlana Sokolova, Mitchell L Leibowitz, Yujun Zhang, Matthew E Hurles, Joshua C Mell, and Ira M Hall. Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. *Genome research*, 20(5):623–635, 2010.

[62] Anne Rovelet-Lecrux, Didier Hannequin, Gregory Raux, Nathalie Le Meur, Annie Laquerrière, Anne Vital, Cécile Dumanchin, Sébastien Feuillette, Alexis Brice, Martine Vercelletto, et al. APP locus duplication causes autosomal dominant early-onset Alzheimer disease with cerebral amyloid angiopathy. *Nature genetics*, 38(1):24–26, 2005.

[63] Scott Schwartz, W James Kent, Arian Smit, Zheng Zhang, Robert Baertsch, Ross C Hardison, David Haussler, and Webb Miller. Human–mouse alignments with BLASTZ. *Genome research*, 13(1):103–107, 2003.

[64] Andrew J Sharp, Ze Cheng, and Evan E Eichler. Structural variation of the human genome. *Annu. Rev. Genomics Hum. Genet.*, 7:407–442, 2006.

[65] Jeremy J Shen and Nancy R Zhang. Change-point model on nonhomogeneous poisson processes with application in copy number profiling by next-generation dna sequencing. *The Annals of Applied Statistics*, 6(2):476–496, 2012.

[66] Suzanne S Sindi, Selim Onal, Luke Peng, Hsin-Ta Wu, and Benjamin J Raphael. An integrative probabilistic model for identification of structural variation in sequencing data. *Genome Biol*, 13(3):R22, 2012.

[67] Pawel Stankiewicz and James R Lupski. Structural variation in the human genome and its role in disease. *Annual review of medicine*, 61:437–455, 2010.

[68] Shin Suzuki, Tomohiro Yasuda, Yuichi Shiraishi, Satoru Miyano, and Masao Nagasaki. ClipCrop: a tool for detecting structural variations with single-base resolution using soft-clipping information. *BMC bioinformatics*, 12(Suppl 14):S7, 2011.

[69] Shu Mei Teo, Yudi Pawitan, Chee Seng Ku, Kee Seng Chia, and Agus Salim. Statistical challenges associated with detecting copy number variations with next-generation sequencing. *Bioinformatics*, 28(21):2711–2718, 2012.

[70] Tom Walsh, Jon M McClellan, Shane E McCarthy, Anjené M Addington, Sarah B Pierce, Greg M Cooper, Alex S Nord, Mary Kusenda, Dheeraj Malhotra, Abhishek Bhandari, et al. Rare structural variants disrupt multiple genes in neurodevelopmental pathways in schizophrenia. *science*, 320(5875):539–543, 2008.

[71] Jianmin Wang, Charles G Mullighan, John Easton, Stefan Roberts, Sue L Heatley, Jing Ma, Michael C Rusch, Ken Chen, Christopher C Harris, Li Ding, et al. CREST maps somatic structural variation in cancer genomes with base-pair resolution. *Nature methods*, 8(8):652–654, 2011.

[72] Zhanyong Wang, Farhad Hormozdiari, Wen-Yun Yang, Eran Halperin, and Eleazar Eskin. CNVeM: copy number variation detection using uncertainty of read mapping. *Journal of Computational Biology*, 20(3):224–236, 2013.

[73] Kris A. Wetterstrand. DNA sequencing costs: Data from the NHGRI genome sequencing program (GSP) available at: www.genome.gov/sequencingcosts. Accessed [November 11, 2013].

[74] David A Wheeler, Maithreyan Srinivasan, Michael Egholm, Yufeng Shen, Lei Chen, Amy McGuire, Wen He, Yi-Ju Chen, Vinod Makhijani, G Thomas Roth, et al. The

complete genome of an individual by massively parallel DNA sequencing. *Nature*, 452(7189):872–876, 2008.

[75] Ruibin Xi, Angela G Hadjipanayis, Lovelace J Luquette, Tae-Min Kim, Eunjung Lee, Jianhua Zhang, Mark D Johnson, Donna M Muzny, David A Wheeler, Richard A Gibbs, et al. Copy number variation detection in whole-genome sequencing data using the Bayesian information criterion. *Proceedings of the National Academy of Sciences*, 108(46):E1128–E1136, 2011.

[76] Ruibin Xi, Semin Lee, and Peter J Park. A survey of copy-number variation detection tools based on high-throughput sequencing data. *Current Protocols in Human Genetics*, pages 7–19, 2012.

[77] Chao Xie and Martti Tammi. CNV-seq, a new method to detect copy number variation using high-throughput sequencing. *BMC bioinformatics*, 10(1):80, 2009.

[78] Xiao Yang, Sriram P Chockalingam, and Srinivas Aluru. A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1):56–66, 2013.

[79] Kai Ye, Marcel H Schulz, Quan Long, Rolf Apweiler, and Zemin Ning. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, 25(21):2865–2871, 2009.

[80] Seungtai Yoon, Zhenyu Xuan, Vladimir Makarov, Kenny Ye, and Jonathan Sebat. Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome research*, 19(9):1586–1592, 2009.

[81] Daniel R Zerbino and Ewan Birney. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome research*, 18(5):821–829, 2008.

[82] Dandan Zhang, Yudong Qian, Nirmala Akula, Ney Alliey-Rodriguez, Jinsong Tang, Elliot S Gershon, Chunyu Liu, et al. Accuracy of cnv detection from gwas data. *PLoS One*, 6(1):e14511, 2011.

[83] Feng Zhang, Wenli Gu, Matthew E Hurles, and James R Lupski. Copy number variation in human health, disease, and evolution. *Annual review of genomics and human genetics*, 10:451–481, 2009.

[84] Qian Zhao, Tao Li, XinZhi Zhao, Ke Huang, Ti Wang, ZhiQiang Li, Jue Ji, Zhen Zeng, Zhao Zhang, Kan Li, et al. Rare CNVs and tag SNPs at 15q11. 2 are associated with schizophrenia in the Han Chinese population. *Schizophrenia bulletin*, 39(3):712–719, 2013.

Curriculum Vitae

Name:	Ehsan Haghshenas
Post-Secondary Education and Degrees:	University of Western Ontario London, ON, CANADA 2012 - 2013 M.Sc. in Computer Science Isfahan University of Technology Isfahan, IRAN 2007 - 2012 B.Sc. in Computer Engineering (Software)
Related Work Experience:	Teaching Assistant Department of Computer Science The University of Western Ontario 2012 - 2013 Teaching Assistant and Lab instructor Department of Electrical and Computer Engineering Isfahan University of Technology 2008 - 2011

Publications:

- N. Asadi, A. Mirzaei, **E. Haghshenas**, "Multiple Observations HMM Learning by Aggregating Ensemble Models", *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5767-5776, 2013
- **E. Haghshenas**, A. Mazaheri, A. Gholipour, M. Tavakoli, "An Intelligent Method for Customizable Adaptive Learning Content Generation", *International Journal of Information and Communication Technology*, vol. 3, no. 1, pp. 43-55, 2011
- N. Asadi, A. Mirzaei, **E. Haghshenas**, "Classification and Clustering of Time Series by Hidden Markov Model Ensembles", *IEEE Transactions on Systems, Man and Cybernetics: Systems*, **(Sbmitted)**