Enabling high-throughput sequencing data analysis with MOSAIK

Author: Michael Peter Stromberg

Boston College Electronic Thesis or Dissertation, 2010

Boston College

The Graduate School of Arts and Sciences

Department of Biology

ENABLING HIGH-THROUGHPUT SEQUENCING DATA ANALYSIS WITH MOSAIK

A dissertation

by

MICHAEL PETER STRÖMBERG

submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

May 2010

Abstract

Enabling high throughput sequencing data analysis with MOSAIK

Michael Peter Strömberg

Dissertation advisor: Gabor T. Marth

During the last few years, numerous new sequencing technologies have emerged that require tools that can process large amounts of read data quickly and accurately. Regardless of the downstream methods used, reference-guided aligners are at the heart of all next-generation analysis studies. I have developed a general reference-guided aligner, MOSAIK, to support all current sequencing technologies (Roche 454, Illumina, Applied Biosystems SOLiD, Helicos, and Sanger capillary). The calibrated alignment qualities calculated by MOSAIK allow the user to fine-tune the alignment accuracy for a given study. MOSAIK is a highly configurable and easy-to-use suite of alignment tools that is used in hundreds of labs worldwide.

MOSAIK is an integral part of our genetic variant discovery pipeline. From SNP and short-INDEL discovery to structural variation discovery, alignment accuracy is an essential requirement and enables our downstream analyses to provide accurate calls. In this thesis, I present three major studies that were formative during the development of MOSAIK and our analysis pipeline. In addition, I present a novel algorithm that identifies mobile element insertions (non-LTR retrotransposons) in the human genome using split-read alignments in MOSAIK.

This algorithm has a low false discovery rate (4.4 %) and enabled our group to be the first to determine the number of mobile elements that differentially occur between any two individuals.

Acknowledgements

During the slow and often interrupted development of my research I accumulated many debts, only a proportion of which I have space to acknowledge here.

I would like to acknowledge the debt I owe to my advisor and friend, Gabor Marth. After joining his lab, he helped transform me from an ambitious programmer into an up-and-coming scientist. His influence in the community has given me opportunities that I would never have imagined prior to beginning my doctoral studies.

I am eternally grateful for the support I have received from my family, especially my father. Throughout my life, he has enthusiastically fostered my computational, academic, and scientific pursuits. I thank my family for always being there for me.

I humbly thank Tony Annunziato for his fortitude in making sure that I lived, breathed, and understood molecular biology.

I am indebted to both Michael Zody and Chip Stewart. My discussions with them concerning data analysis and statistical modeling have had a profound effect on how I perceive challenging problems in computational biology.

I would like to thank all of the users and sequencing workshop students for testing my software. All of the great features in MOSAIK originated from someone with a “crazy idea”.

Thank you all.

Figures

Tables

1. Introduction

1.1. Background to DNA research

1.1.1. DNA: The primary source of genetic information

In the early 20th century, genes were believed to be substance-less entities and proteins were suspected of being able to pass on genetic information[1]. In 1910, Thomas H. Morgan's fruit fly (*D. melanogaster*) research at Columbia University revealed that genes were carried on specific chromosomes[2]. A few decades later, a team of medical scientists led by Oswald T. Avery were the first to show that isolated deoxyribonucleic acid (DNA) was responsible for transforming non-encapsulated variants of pneumococcus into encapsulated cells[3] and therefore demonstrated that DNA was the primary source of genetic information.

In April 1953, three papers suggested a strong hypothesis for the alpha-helical structure of DNA[4-6]. Rosalind Franklin obtained x-ray crystallographic photos of the DNA molecule and was the first to conclude that DNA consisted of two chains of nucleotides. Five years later, Francis Crick presented the central dogma of molecular biology[7] (Figure 1.1). Shortly after this formulation, the degenerative, non-overlapping nature of how DNA triplets translate into amino acids was elucidated[8,9].

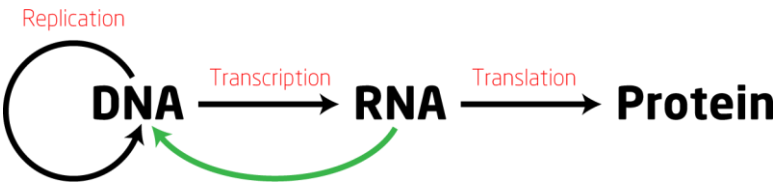

Figure 1.1. The central dogma in molecular biology states that DNA can be transcribed into RNA which in turn can be translated into a protein. Finally, DNA can be replicated with a protein called DNA polymerase and few other accessory proteins.

With these early events in nucleic acid research; development of discoveries in cloning vectors[10], polymerase chain reaction (PCR)[11,12], and sequencing methods[13,14] have largely enabled the modern field of genomics.

1.1.2. The genomics era

The Human Genome Project was an ambitious undertaking where the human genome and genomes of five other model organisms (*Escherichia coli*, *Saccharomyces cerevisiae*, *Drosophila melanogaster*, *Caenorhabditis elegans*, and *Mus musculus*) were sequenced and assembled into reference sequences[15]. Besides improving the resources available to molecular biologists, sequencing model organisms served as a methods development study that would determine the methods used when decoding the full human genome. In an effort that included more than 20 sequencing centers in six countries, a draft of the human genome was already publicly available in 2000[16] and the complete genome was made available in 2003[17,18]. More than 20,000 bacterial artificial chromosome (BAC) clones of approximately 160 kb were produced from segments of DNA inserted from the human genome[19]. The BAC clones were amplified in bacterial culture, isolated in large quantities, and then sheared into 2 – 3 kb fragments. The fragments were then subcloned into plasmid vectors and amplified once again in a bacterial culture. The DNA was extracted and sequenced using gel-based Sanger dideoxy sequencing[14,20]. The sequences were then assembled *in silico* into contiguous consensus sequences. A finishing process was used to fill in the gaps between the consensus sequences.

The genomics field has grown substantially since the Human Genome Project. As of September 2009, the Genomes Online Database[21] (GOLD) indicates that 1,095 genomes have been completely sequenced and 4,543 genomes are currently being sequenced. With the goal of rapidly releasing sequence assemblies and sequencing data to the public, the data available to the research community is growing exponentially every year[15]. With the increasing amount of sequencing data available and the relatively low utility of raw genome sequence[22], the focus of genome sequencing has shifted from reconstructing the reference sequence toward downstream analysis projects. This shift in focus has led to the development of emerging fields such as comparative genomics, personal genomics, metagenomics, and epigenomics.

1.2. Current DNA sequencing technologies

By the end of the Human Genome Project, Sanger capillary dideoxy sequencing[15] was the best sequencing technology available. Since then, two new generations of sequencing technologies have emerged that are cheaper and faster, thereby allowing studies and methodologies that were not feasible with the throughput attainable from Sanger sequencing[23]. The increase in throughput in turn increases the demand for faster reference-guided alignment and *de novo* assembly tools.

1.2.1. First-generation sequencing technologies

Sanger capillary

Sanger dideoxy sequencing[14,20] was used throughout the Human Genome Project. While the first automated Sanger sequencing machine was released by Applied Biosystems back in 1987, the most recent iteration of machines (Sanger capillary) became available in 1999 and produce 96 reads with an average read length of 700 bp (1.6 Mb per run). The estimated price for 1x coverage of the human genome is around $1.4M (Table 1.1)[24]. Roughly 7x coverage is required to successfully create a *de novo* assembly of the human genome using Sanger capillary sequences[16].

In Sanger dye-terminator sequencing, a single-stranded DNA template (primer) is used to guide the DNA polymerase. The polymerase incorporates the appropriate deoxynucleotides (dATP, dCTP, dGTP, dTTP) until a fluorescently labeled chain-terminating dideoxynucleoside triphosphate is incorporated. The DNA fragments are then denatured and separated based on length by capillary electrophoresis. During electrophoresis the emission wavelengths of the fluorescent labels are detected.

1.2.2. Second-generation sequencing technologies

The second generation of sequencing technologies became available in 2005, when the Roche 454 pyrosequencer became available. Instead of amplifying DNA by time-consuming bacterial cloning, PCR amplification is used in this generation of

sequencing technologies: Applied Biosystems SOLiD, Complete Genomics, Illumina, and Roche 454.

Roche 454

The GS FLX Titanium series represents the third iteration of the Roche 454 pyrosequencing platform[25] which was originally released in 2005. Of the currently available second and third generation sequencing technologies, the Roche 454 platform offers the longest read lengths, averaging 400 bp. Each run takes roughly 10 hours and produces between 400 – 600 Mb of read data. The estimated price for 1x coverage of the human genome is around $143k (Table 1.1)[24].

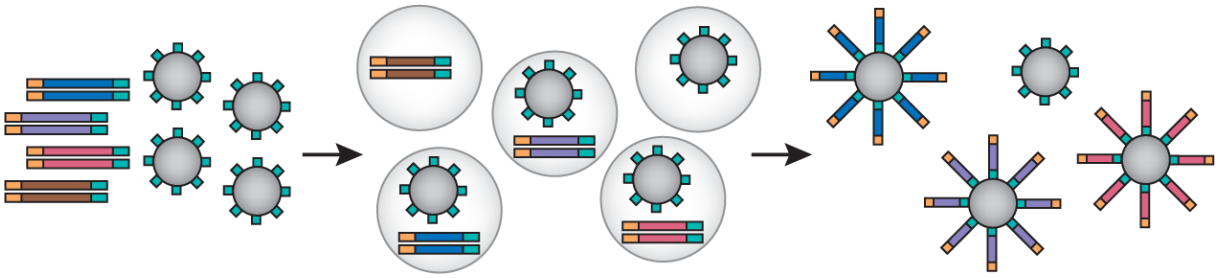

Figure 1.2. Emulsion PCR. Fragments with adaptors (gold and turquoise) are multi-template PCR amplified within a water-in-oil emulsion. The 5' primer is tethered to the surface of a bead. Beads with attached PCR amplicons can be selectively enriched. Reprinted from [26].

DNA fragments are ligated with 454-specific adapter sequences and mixed with agarose beads that contain surface oligos that are complementary to the adapter sequence. Emulsion PCR is used to attain one million fragments on each agarose bead (Figure 1.2). The agarose beads are added to wells in the picotiter plate (PTP). Each well is physically dimensioned to accommodate only one bead. Enzyme-containing beads are added to the PTP and centrifuged to surround and lock in the agarose beads. Pure nucleotide solutions are stepwise introduced to the PTP in a

predefined incorporation order (TACG). Incorporation of the nucleotide by the DNA polymerase releases pyrophosphatase (PPi). ATP sulfurylase converts the PPi into ATP, which activates the luciferase-based light output. The amount of light produced is proportional to the number incorporated nucleotides; however homopolymer runs longer than 6 bp cannot be detected accurately (Figure 1.3). The Roche 454 platform exhibits a low error rate (Figure 1.4) that is dominated by insertion and deletion errors (Figure 1.5) caused by the difficulty in determining the number of incorporated bases.

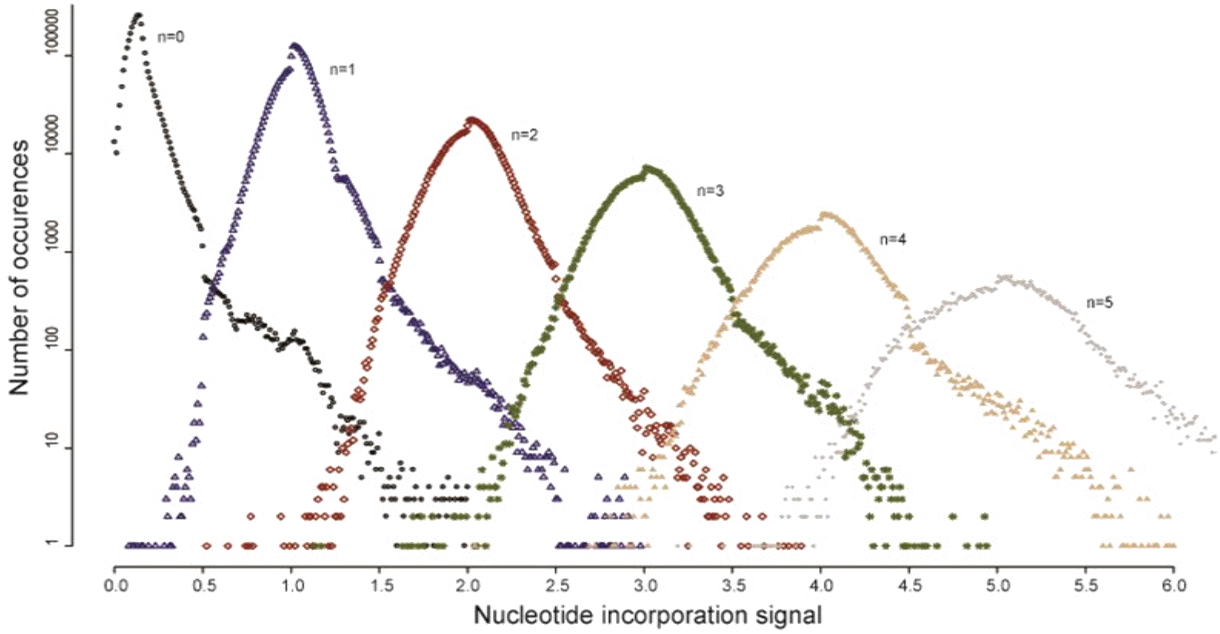

Figure 1.3. A distribution of nucleotide incorporation signals is shown for known homopolymers of lengths between 0 bp and 5 bp. The variance of each distribution grows larger as the homopolymer length increases. Due the large variance, it is difficult to determine the correct homopolymer length accurately based on the nucleotide incorporation signal. Reprinted from unpublished research by Aaron Quinlan (Boston College).

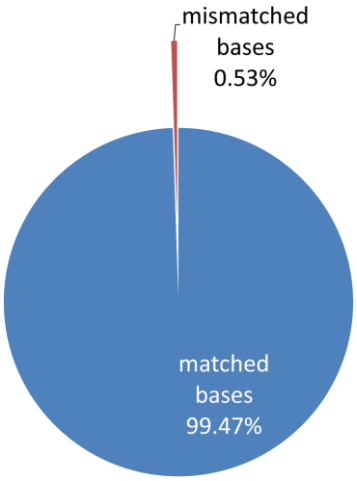

Figure 1.4. Mismatched bases were quantified in a Roche 454 data set. Some mismatched bases may reflect genetic variants, but most represent sequencing errors. Adapted from unpublished research by Derek Barnett (Boston College).

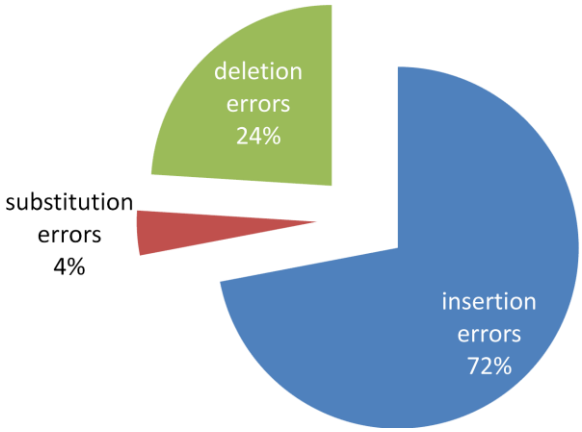

Figure 1.5. Sequencing error breakdown. Of the mismatched bases in Figure 1.4, 72 % represent insertion errors and 24 % represent deletion errors. Only 4 % of the mismatched bases reflect substitution errors. Adapted from unpublished research by Derek Barnett (Boston College).

Illumina

Illumina[27] recently released their third iteration (HiSeq 2000) of their sequencing by synthesis platform which was originally released in 2005. A HiSeq 2000 run produces up to 200 Gb of high quality reads in approximately eight days. Read lengths vary from about 35 bp to 100 bp. The estimated price for 1x coverage of the human genome is around $8k (Table 1.1).

DNA samples are fragmented and ligated with Illumina-specific adaptors. Using the cluster station, these fragments bind with the oligonucleotide surface of a flow cell and undergo bridge amplification (Figure 1.6) by DNA polymerase to produce clusters. Approximately one million copies of each fragment are required to achieve the necessary signal intensity during sequencing. All four nucleotides are added simultaneously to the flow cell. The DNA polymerase incorporates a fluorescently labeled nucleotide and a 3′-OH group is chemically blocked to prevent

homopolymer incorporation. After incorporation, imaging is performed in three 100-tile segments where each tile contains approximately 30,000 clusters. The 3′-OH blocking group is removed and another sequencing cycle begins.

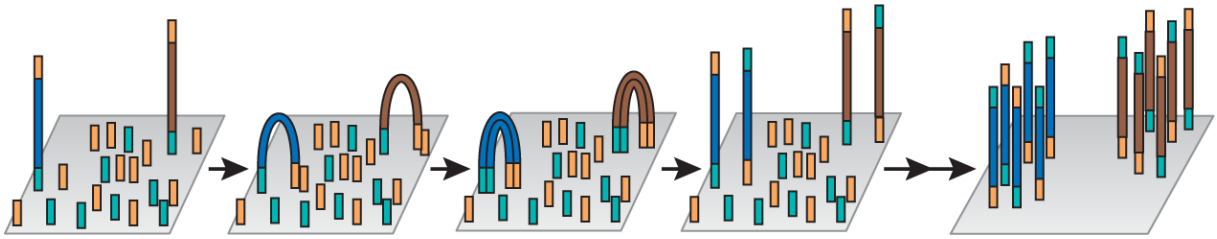

Figure 1.6. Bridge amplification. DNA fragments are flanked by adaptors and bound to a surface coated with two types of primers, corresponding to the adaptors. Amplification occurs iteratively with one end of each bridge tethered to the surface. Reprinted from [26].

After each incorporation cycle, there is a probability that some DNA fragments will be out of phase with the rest of the cluster. Some fragments will lag behind (phasing) and some will be ahead (pre-phasing) of the current incorporation cycle. Due to the effects of phasing and pre-phasing, sequencing errors tend to accumulate at the end of the reads. The Illumina platform exhibits a low error rate (Figure 1.7) that is dominated by substitution errors (Figure 1.8), which are most likely due to the aforementioned pre-phasing and phasing artifacts.

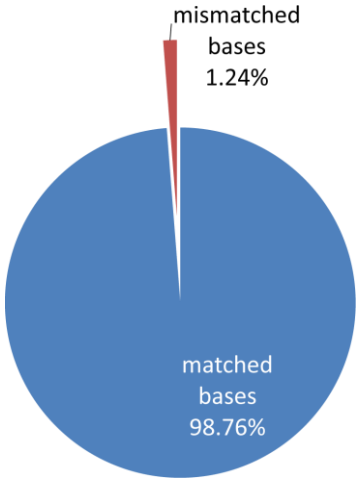

Figure 1.7. Mismatched bases were quantified in an Illumina 36 bp data set. Some mismatched bases may reflect genetic variants, but most represent sequencing errors. Adapted from unpublished research by Derek Barnett (Boston College).

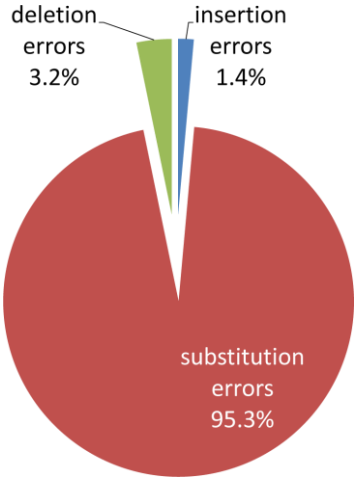

Figure 1.8. Sequencing error breakdown. Of the mismatched bases in Figure 1.7, 95 % represent substitution errors. Only 3 % of the mismatched bases reflect deletion errors and 1 % reflect insertion errors. Adapted from unpublished research by Derek Barnett (Boston College).

Applied Biosystems SOLiD

The Applied Biosystems SOLiD sequencing platform uses sequencing by ligation and exhibits the unusual characteristic of sequencing base to base transitions instead of the actual bases[28]. The current iteration of the SOLiD platform (SOLiD 4) allows up to two flow cells per run with up to eight individual samples in each flow cell. Up to 100 Gb of aligned sequence data can be produced with standard 50 bp mate-pair runs. The estimated price for 1x coverage of the human genome is around $8k (Table 1.1).

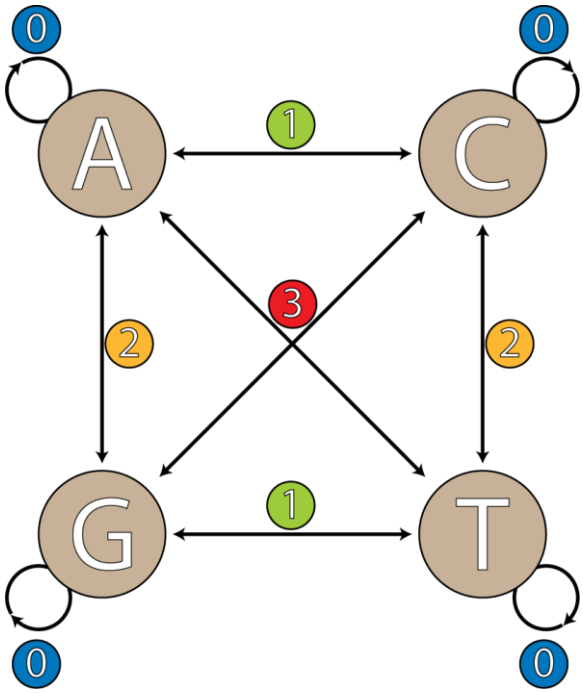

Figure 1.9. Basespace to color space finite state automaton. Despite the term color space, the dibase transitions are usually encoded as numbers. For example, transitions from a base to itself are encoded as 0 and a transition from A to T would be encoded as 3.

The sample preparation for the SOLiD platform involves amplifying an adapter-ligated fragment library with emulsion PCR on 1 µm magnetic beads. Subsequently, primers are annealed to shared adapter sequences on each fragment. DNA ligase is used to anneal semi-degenerate 8-mer oligonucleotides to the universal sequence primer where the first five bases complement the template sequence and the first two bases are labeled with a fluorescent dye. Four different fluorescent dyes encode the sixteen possible two base combinations (Figure 1.9). The dye is then detected by an imaging stage, the unextended strands are capped with phosphatase, and the last three bases are cleaved off[29]. This process is repeated for five more extension rounds while varying the position of the universal sequence primer (Figure 1.10).

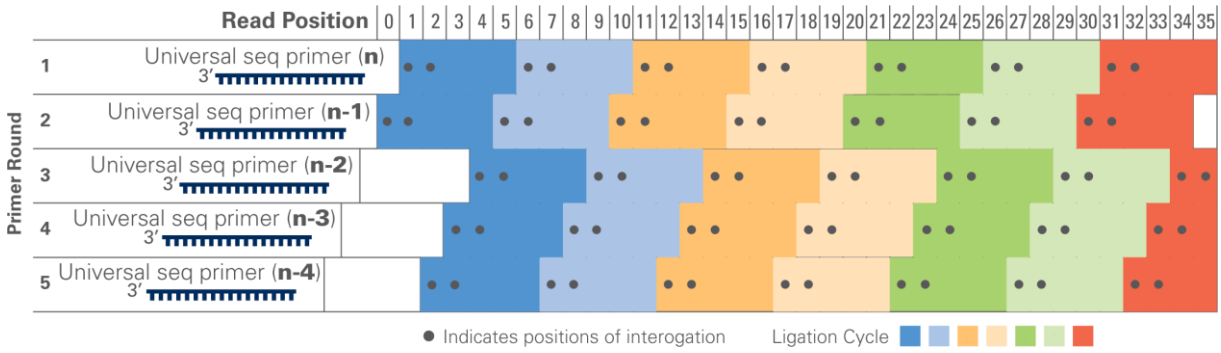

Figure 1.10. Five primer rounds are used interrogate each internal read position twice. Each primer round interrogates two consecutive bases in the read. Reprinted from [29].

Complete Genomics

In contrast to the other sequencing technologies, Complete Genomics[24] has chosen a service-oriented model rather than selling machines to customers. Due to the service-oriented model, it is still unclear if read data will be available to customers who wish to use alternative aligners. Complete Genomics is mentioned in the interest of completeness, but remains beyond the scope of the aligner development and analysis studies mentioned later in this thesis. Since most molecular biology labs send samples out for Sanger sequencing, it seems plausible that labs will want to do the same with more modern sequencing technologies. At the moment, they have delivered 50 genomes to customers, an additional 500 have been ordered, and they hope to ramp up to sequencing 500 genomes per month by the end of 2010. By improving the sequencing yield, building 20 data centers around the world, and adding more sequencing machines at each data center; their aim is to sequence one million genomes within the next five years.

Each run currently takes about 11 days and produces around 18 genomes per run (2 Tb). The read lengths are adequate at 35 bp, but a bit short when compared

with improvements in the Illumina and AB SOLiD platforms. A server farm consisting of 1.5 PB of storage and 6500 processor cores handles the downstream bioinformatics such as read alignment and genetic variant calling. They estimate their false discovery rate for genetic variant calling to be around 0.2 %. The estimated price for 1x coverage of the human genome is around $110 (Table 1.1)[24].

The sample preparation for the Complete Genomic platform involves fragmenting the DNA and recursively cutting the fragments with type IIs restriction enzymes. A directional adapter is inserted and the resulting circles are replicated with Phi29 polymerase. DNA nanoballs (DNBs) are formed from hundreds of copies of sequencing substrate in palindrome-promoted coils of singled-strand DNA (ssDNA) and are bound to the surface silicon substrate. Combinatorial probe anchor ligation (cPAL) sequencing chemistry is used to independently read up to 10 bp adjacent to each of the eight anchor sites[28,30,31]. A degenerate oligonucleotide probe with a fluorescent dye confirms the base at a given interrogation position and favors ligation if the probe is complementary (Figure 1.11). Four different fluorophores indicate the base at a given interrogation position.

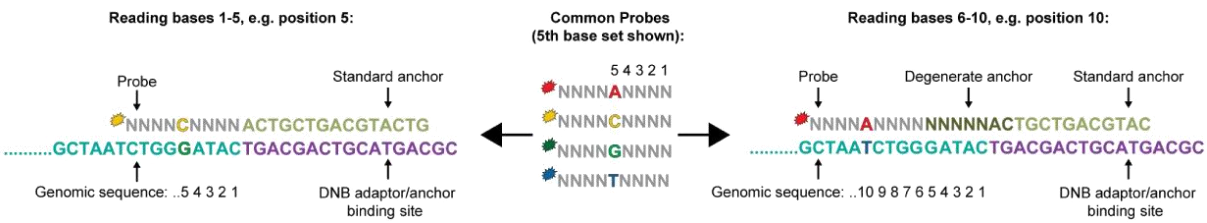

Figure 1.11. Degenerate oligonucleotides are used to interrogate bases at eight different anchor positions. Two anchor positions are shown in this figure, with a set of probes that interrogate the 5th base. Reprinted from [24].

Due to the unchained sequencing chemistry, the bases are sequenced in a stochastic and independent manner. Therefore sequencing errors are not propagated as the read length increases, as is common with other technologies that use sequencing by ligation or sequencing by synthesis.

1.2.3. Third-generation sequencing technologies

The general trait common to the third generation of sequencing technologies is single molecule sequencing. This eliminates the need to amplify DNA during sample preparation. Helicos is the only third generation sequencing technology currently available, but single molecule sequencing platforms are currently being developed by Applied Biosystems, Illumina, and Pacific BioSciences.

Helicos

The PCR amplification used in most current sequencing technologies is problematic since amplification efficiency varies as a function of template properties, introduces errors, and introduces uncontrolled bias in template representation. Since Helicos[32] is a single molecule sequencing platform, the problems associated with PCR amplification can be avoided. It takes the machine eight days to process two flow cells simultaneously (6 days to process one flow cell) producing around 21 – 28 Gb of sequence per run. The read lengths vary from 25 bp to 55 bp with an average read length hovering around 30 bp to 35 bp[32]. Roughly 0.2 % of the bases are substitution sequencing errors while insertion and deletion sequencing errors occur more

frequently (1.5 % and 3.0 % respectively). The estimated price for 1x coverage of the human genome is around $2k (Table 1.1)[24].

Helicos uses a DNA polymerase to sequence by synthesis. Poly(dT) oligonucleotides are covalently anchored to glass cover slips. These oligos capture single stranded, poly(dA)-tailed templates and act as a primer for the template-directed primer extension. Up to 224 sequencing cycles are performed and during each cycle, polymerase and labeled nucleotides are added, the excess is rinsed away, imaging is performed, and finally the dye labels are cleaved away. To reduce the error rate, the sequenced template can be melted off using hot water and then the templates can be primed again for another sequencing pass.

1.2.4. Price per 1x coverage of the human genome

Table 1.1. Sequencing cost. Even though these prices reflect the cost for 1x coverage of the human genome, it is important to note that sequencing technologies that use shorter read lengths normally require deeper coverage for *de novo* assembly and genetic variant calling than sequencing technologies with long read lengths[24]. This is also discussed in the *Pichia stipitis* mutational profiling study (Section 3.2).

	Cost per 1x
Sanger capillary	$1.4M
Roche 454	$143k
Illumina	$8k
Applied Biosystems SOLiD	$8k
Complete Genomics	$110
Helicos Heliscope	$2k

1.2.5. Using large sequence fragments with short-read technologies

In general, long reads can be aligned to a larger portion of the genome than short reads (discussed in Section 1.3.3). Since many of the newer sequencing technologies

have short read lengths, two methods (paired-end and mate-pair sequencing) can be used to sequence both ends of a larger fragment. Paired-end sequencing involves repairing the ends after fragmentation, adding an adenine nucleotide to the 3′ end, and ligating vendor-specific paired-end adaptors (Figure 1.12). Mate-pair sequencing involves repairing the ends after fragmentation, labeling the ends with biotin and circularizing the fragment, and fragmenting the circularized fragment. The fragments containing biotin are pulled using streptavidin beads, the ends are repaired, and vendor-specific mate-pair adaptors are ligated to the ends (Figure 1.13).

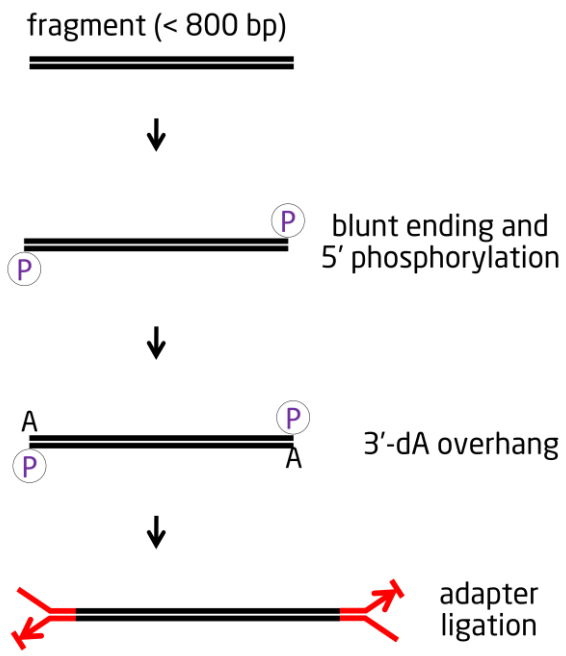

Figure 1.12. Paired-end library sequencing preparation.

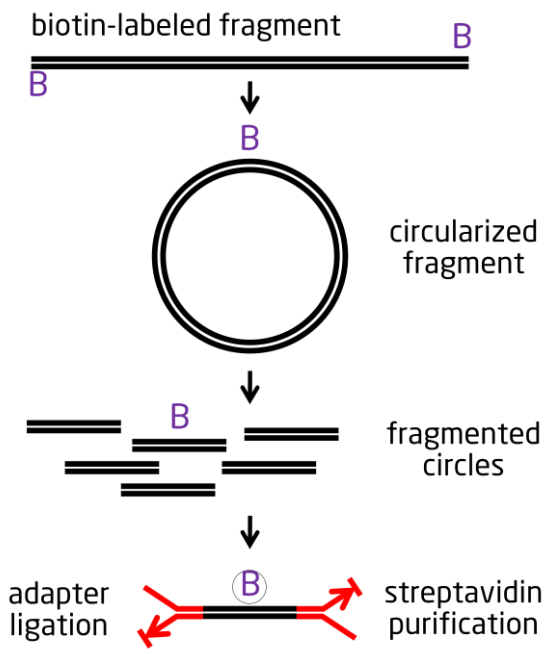

Figure 1.13. Mate-pair library sequencing preparation.

The ends of the library fragments (mates) are separated by a known distance that is determined by gel size selection and results in a fragment length distribution. The mates on each end of the fragment usually have opposite orientations in paired-

end sequencing and usually have the same orientation in mate-pair sequencing (Figure 1.14). Read-pair sequences can be aligned back to the genome almost as accurately as longer sequences using these restrictions, while benefiting from the high throughput of short-read technologies.

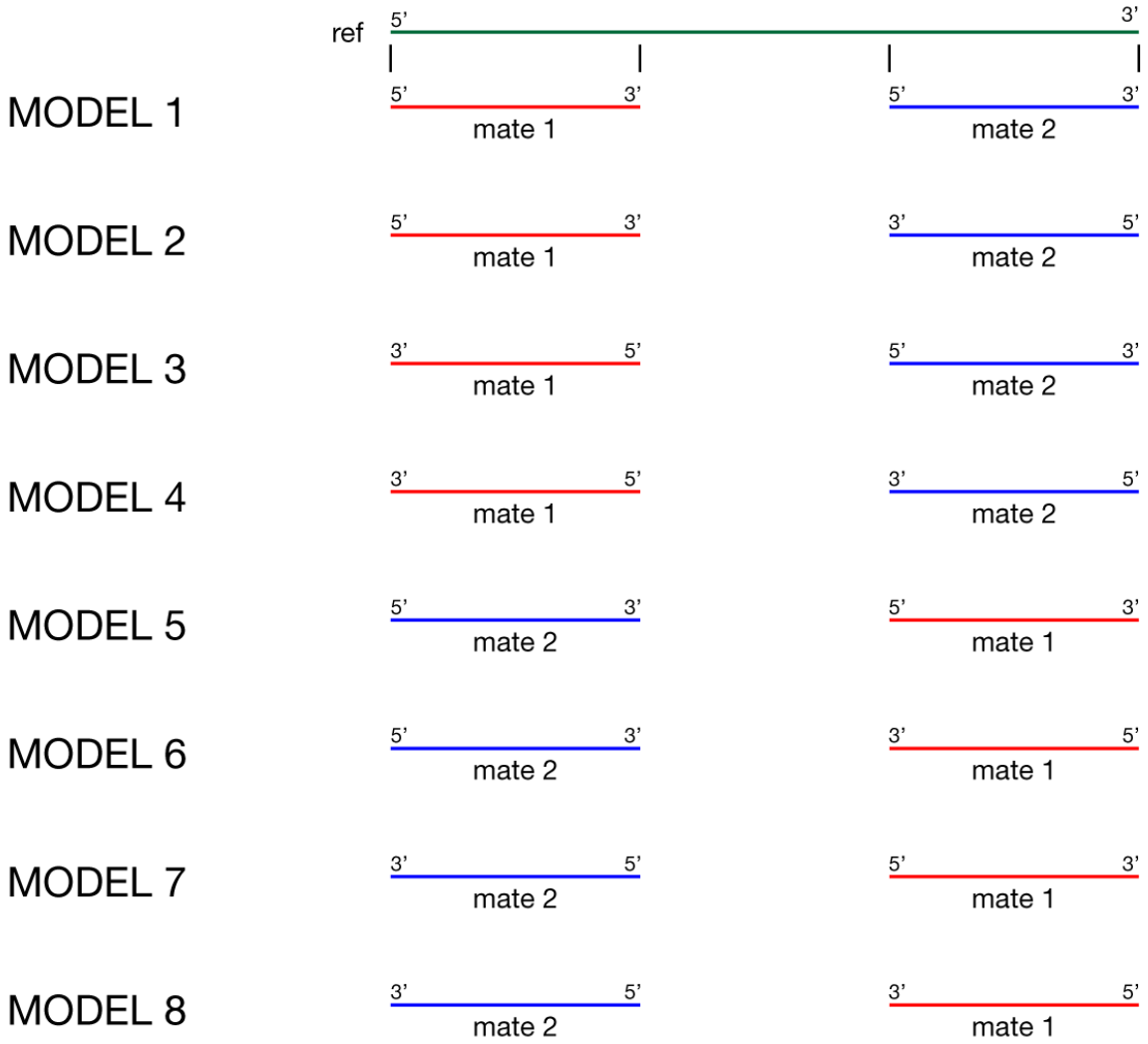

Figure 1.14. All possible combinations of mate order and orientation. Illumina paired-end reads typically conform to models 2 and 6. AB SOLiD and Roche 454 mate-pair reads typically conform to models 4 and 5.

1.3. Alignment algorithms

Alignment algorithms fall into two categories: *de novo* assembly and reference-guided alignment algorithms. *De novo* assemblers use reads to reproduce a reference

sequence, whereas reference-guided algorithms align the reads to an existing reference sequence.

1.3.1. *De novo* assembly

A healthy rivalry existed between the public Human Genome Project and the private effort led by Craig Venter at Celera Genomics[33]. It concerned which team could be the first to assemble and publish the human genome reference sequence. The Human Genome Project employed a hierarchical shotgun sequencing strategy, which meant that contigs were generated from BAC clones. Shotgun sequencing was in turn used on these contigs, generating several million reads. Jim Kent's GigAssembler[34] openly competed with the Celera Assembler[35]. GigAssembler was the first to produce an assembled human genome reference sequence although both programs finished within three days of each other.

Like most *de novo* assembly programs at the time[36-39], both of these assemblers implemented an overlap-layout consensus algorithm. In computational complexity theory, this algorithm can be reduced to a Hamiltonian path problem that is solved in nondeterministic polynomial time, but can be verified in polynomial time (NP-complete)[40]. These assemblers compute the overlaps between reads, and unique overlaps are identified and assembled into contigs. From these contigs, a multiple sequence alignment is constructed and used to create a consensus sequence. The assemblers differ mainly in how they handle potential sources of assembly artifacts.

Poor end regions in Sanger reads, chimeric and repetitive regions in the genome, and false overlaps are some common sources of assembly artifacts.

Instead of using the overlap-layout consensus algorithm, newer assemblers use graph or tree data structures to solve the *de novo* assembly problem[40-45]. These algorithms belong to a much simpler complexity class and can often be performed within O(n log n) time[41]. This class of *de novo* assembly algorithms is therefore especially attractive when working with newer sequencing technologies that offer orders of magnitude more reads than previous Sanger capillary technology. These assemblers divide the reads up into small hashes and assign these hashes to nodes in the graph. These graphs are then simplified and curated to remove the effects of sequencing errors. These assemblers typically differ on the underlying data structure used (*e.g.* de Bruijn graph, Eulerian graph, or prefix tree) and how they handle potential sources of assembly artifacts.

1.3.2. Reference-guided alignment

Reference-guided alignment involves aligning a set of reads to one or more reference sequences. Most aligners divide the reads up into one or more hashes, use those hashes to find associated reference sequence locations, use the sequence locations to determine where potential alignment may occur, and perform pairwise alignment between the read and that region of the reference genome[46-48]. Occasionally it is convenient with respect to memory usage to swap the role of the reference sequences and the reads in the above method[49]. Some of the newest aligners use the Burrows-

Wheeler transform to produce a compressed suffix array that can be quickly traversed with each read to find the associated reference sequence locations[50-53].

1.3.3. Challenges when aligning short reads

Unique hashes in the genome

Many de novo assemblers and aligners split reads into small k-mers (hashes) to correctly align reads that contain sequencing errors and genetic variants. At a hash size of 15, only 32.4 % of the hashes are unique in the human genome (Table 1.2). The unique 15 bp hashes, however, only address 6.2 % of the genome (Figure 1.15). When the hash size is increased to 32 bp, 97.4 % of the hashes are unique and address 86.6 % of the human genome. This gives an optimistic estimate with respect to how much of the genome can be covered by short 32 – 35 bp reads. When factoring in the number of sequencing errors and the role of microrepeats (discussed in Section 3.1.2), the expected percentage of the covered genome decreases.

Table 1.2. Unique human genome coverage using exact matches (hashes)

	11 bp	15 bp	17 bp	19 bp	32 bp
# of locations per hash (mean)	684	5.25	1.71	1.31	1.12
% unique hashes	0.1 %	32.4 %	76.9 %	93.0 %	97.4 %
% unique genome coverage	0.0 %	6.2 %	45.1 %	70.7 %	86.6 %

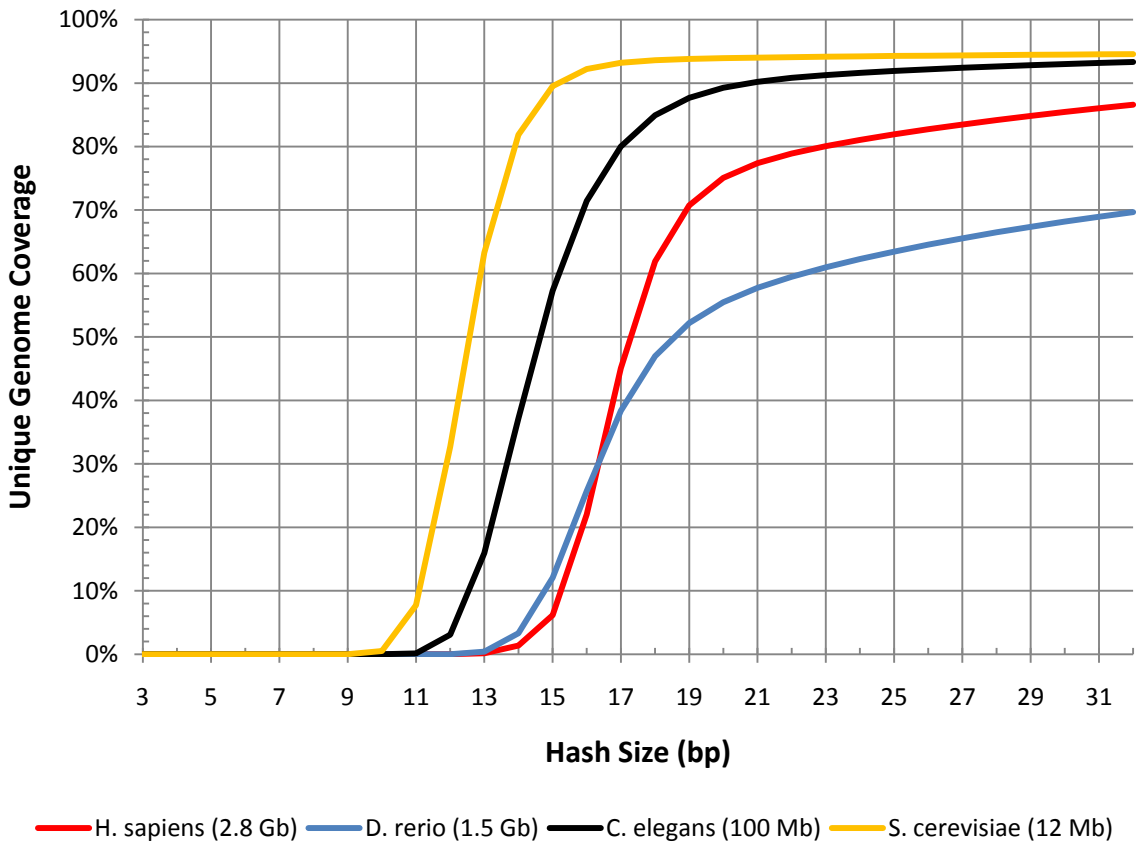

Figure 1.15. Unique genome coverage with respect to increasing hash size. Four model organisms are shown: humans (red), zebrafish (blue), roundworm (black), and yeast (orange). In general, organisms with smaller genomes reach a higher percentage of unique genome coverage at lower hash sizes. However, high repeat content (zebrafish) has a detrimental effect on the percentage of unique genome coverage.

BLAT and BLAST are poorly suited for short read alignment

Two of the most popular reference-guided aligners, BLAT[46] and BLAST[54], were designed with longer, Sanger reads in mind. Due to this design decision, they used non-overlapping hashes to seed alignments. Even when these programs are forced to use smaller, overlapping hashes to seed short read alignments, the alignment performance is prohibitively slow. With these sensitive settings, only the significant hash matches are used when seeding an alignment rather than using all the hash matches. As a result, BLAT and BLAST are poorly suited for aligning short reads.

Seed-and-extend alignment algorithms

Alignment speed is critical when trying to process millions of short reads in a timely manner. To improve alignment speed, most aligners resort to faster, ungapped alignment algorithms that use seed-and-extend alignment algorithms. In these cases, an alignment is seeded by hash and then the alignment is extended in both directions while sequence identity remains high over a predefined threshold. The resulting ungapped alignments lead to artifacts when biological insertions and deletions (INDELs) occur. To fix these artifacts, utilities have been created to locally reassemble the regions where INDELs are believed to occur[55].

Optimal alignment algorithms

Only a couple of short read aligners use dynamic programming algorithms such as the Smith-Waterman[56] or Needleman-Wunsch[57] algorithms that are guaranteed to return the optimal alignment given the current scoring scheme. The Smith-Waterman algorithm provides a gapped local pairwise alignment while the Needleman-Wunsch algorithm provides a gapped global pairwise alignment. Global alignments are aligned across the entire read. Local alignments are aligned across the best matching region in the read. Gapped algorithms are much better than ungapped algorithms at discovering short-INDELs (discussed later in Section 3.3.4: The Wellcome Trust Sanger Institute (WTSI) used an ungapped alignment algorithm[49], I used a gapped alignment algorithm).

Illumina and AB SOLiD reads tend to accumulate sequencing errors at the ends of the reads. For these technologies, it is often beneficial to use a local alignment

algorithm to align the 5′ region in the read where the probability of sequencing errors is lower. If a global alignment algorithm is used, the entire read will be aligned. In this case, sequencing errors included in the 3′ region may complicate downstream genetic variant calling.

1.3.4. Output formats

A common problem in bioinformatics is that the output format produced by one program is often not in the input format required by the next program in the analysis pipeline. To cope with this problem, bioinformaticians are required to write programs that convert between the two formats. When Sanger capillary reads represented the latest sequencing technology, the ace file format produced by phrap[39] was the *de facto* assembly file format and was supported by many SNP discovery and visualization tools. When the field transitioned to using the newer technologies and reference-guided aligners, a *de facto* file format did not exist for several years. Fortunately, a standard alignment format (SAM/BAM) was defined and published in early 2009[58]. The SAM format is the text representation of the alignment format and the BAM format is the binary representation of the alignment format. With increased adoption of the SAM/BAM format, it will be easier for users to specify exactly which aligner and downstream analysis tools they wish to use in their pipelines.

1.4. Genetic variant discovery by DNA sequencing

1.4.1. Single nucleotide and short insertion-deletion polymorphisms

During the past several years, single nucleotide polymorphism (SNP) discovery tools have transitioned from using filter-based heuristics[59] to using more sensitive probabilistic or machine learning approaches[49,60-64]. Most of the probabilistic solutions use the alleles and associated base qualities at a given reference position to calculate the consensus genotype and the posterior probability the consensus genotype is correct. The individual genotypes are then used to calculate a posterior probability, P(SNP), that the reference position is polymorphic with respect to the sample population. A SNP candidate is called if P(SNP) is greater than a user-specified threshold. The machine learning approaches generally use features such as P(SNP), sequence depth, major and minor allele frequency, maximum and average qualities for the major and minor alleles, and alignment quality to decide if a locus is polymorphic[62,65].

An abundance of SNP discovery tools exist, but only two of the tools currently available attempt to find short-insertion and deletion (INDEL) polymorphisms[49,60]. PolyBayes[60] treats the gaps in the reads and reference sequence as a fifth nucleotide where the base quality represents the minimum of the two flanking non-gapped bases. It is unclear how the SNP calling algorithm implemented in MAQ[49] treats INDELs.

Filters are frequently applied to the resulting SNP candidates. Thresholds for P(SNP), aligned read coverage, SNP-to-SNP distance, and the Hardy-Weinberg test statistic have been used to screen potential false positives. To effectively establish these thresholds, a baseline is created using the subset of SNP candidates that overlap with confirmed variants (such as the HapMap3 genotypes[66]). Applying the same filter to the remaining SNP candidates can be used to help determine the appropriate minimum and maximum filter values.

Previous studies have indicated that a 2:1 transition to transversion (ti:tv) ratio is typical of human SNPs[67,68] and in mammalian evolution[69]. By calculating the ti:tv ratio when calling SNP candidates in humans, SNP calling and filtering parameters can be fine-tuned until the ti:tv ratio converges to 2.0.

1.4.2. Structural variation detection

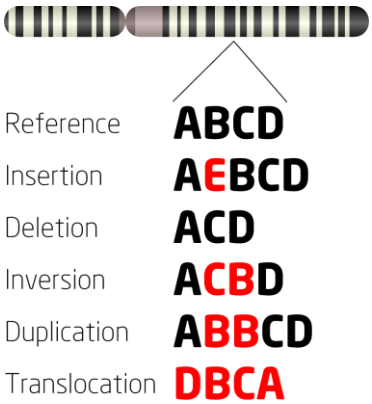

Figure 1.16. Classification of structural variants with respect to the reference sequence. Adapted from [71].

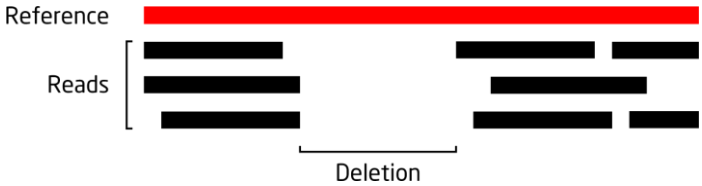

Figure 1.17. Read coverage algorithm. Reads (black) are aligned to the reference sequence (red). The reference sequence has relatively equal read coverage everywhere except for the deleted region.

Structural variations are considered to be genomic alterations that are longer than 1 kb and are commonly classified into the following types: insertions, deletions,

inversions, duplications, and translocations[70,71] (Figure 1.16). Structural variation discovery tools often use read coverage, read-pair fragment length distributions, and split-read analysis to identify structural variation candidates. Read coverage (Figure 1.17) and read-pair algorithms (Figure 1.18) detect deviations in the read coverage and fragment lengths of paired-end reads respectively to determine structural variants. Split-read algorithms (Figure 1.19) detect structural variations by partially aligning the read to two disparate locations. Programs such as BreakDancer[72] and Spanner[73] use two algorithms, read coverage and read-pair, to increase specificity. The remaining structural variation tools that are currently available use the read-pair algorithm[74-76].

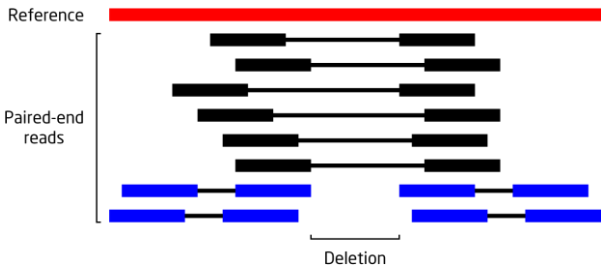

Figure 1.18. Read-pair algorithm. Concordant paired-end reads (blue) have a smaller fragment length than the discordant paired-end reads (black) that span the deletion in the reference. None of the individual mate sequences in the paired-end reads align to the deleted region.

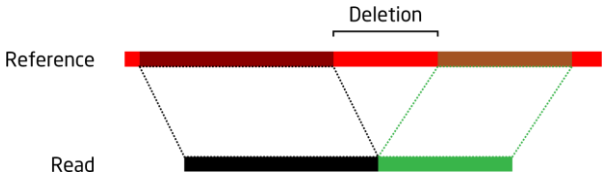

Figure 1.19. Split-read algorithm. One read is aligned to the reference sequence (red). The 5′ end of the read (black) aligns to one part of the reference and the 3′ end of the read (green) aligns to another part of the reference. The split-read spans a deletion in the reference.

1.5. Research focus

The core focus of the Marth Lab at Boston College is genetic variant discovery. During the last few years numerous new sequencing technologies have emerged that require tools that can process large amounts of read data quickly and accurately. After analyzing the initial batch of second-generation sequencing data sets in 2006, a

decision was made to create a highly configurable reference-guided aligner rather than using the two aligners available from the respective sequencing companies.

This thesis has been divided into four main sections. This first section focuses primarily on the design and development of our aligner, MOSAIK. After each round of iterative development, MOSAIK has been applied to several genetic variant analysis projects – each successively larger than the previous one. The second section focuses on four of the genetic variant analysis projects that affected MOSAIK development the most. One of the true tests of an aligner is the capability to apply it to an unexpected scenario without needing to modify the underlying codebase. This was the case in the third section in which MOSAIK was used to discover mobile element insertions in the human genome using split-read alignments. The final section of thesis discusses future avenues of development and how the sequencing landscape may change in the next few years.

2. MOSAIK

2.1. Introduction

The widespread availability of second-generation sequencing platforms has enabled cheaper resequencing efforts[24] with ultra-high throughput. Sequencing technologies such as Illumina, Roche 454, Complete Genomics, and Applied Biosystems SOLiD have been driving the current market, whereas Pacific Biosciences SMRT[77] and the Helicos Heliscope are leading the development of third-generation of sequencing instruments. The flexibility and throughput of these technologies are enabling research in genetic variant discovery, epigenomic variation discovery, RNA-Seq, and ChIP-Seq[23]. As the sequencing throughput improves, alignment and analysis pipelines will also need to keep pace.

Each of these sequencing technologies has a different error model, base quality assignment algorithm, and range of read lengths. Besides presenting data in basespace, some instruments present the data in different representations such as the dibase encoding (color space) used in SOLiD reads. Single-molecule sequencing technologies sometimes exhibit bases that have not been registered by the instrument (dark bases)[32]. It is a challenge to create a reference-guided aligner that not only handles several sequencing technologies, but also deals with the caveats associated with each technology.

In addition to speed and accuracy, there are many traits that are desired in aligners. Aligners that produce gapped-alignments produce fewer alignment

artifacts than ungapped aligners and enable short-INDEL discovery. Finding all the possible alignment locations for a given read is essential in applications such as copy number and structural variation discovery. Aligners that support many different sequencing technologies and store output in standard formats such as SAM and BAM[58] are more likely to be used in production pipelines. Highly configurable aligners enable users to experiment with new application areas. Finally, an aligner should provide a probability that a particular alignment is misaligned so the user can weight the data accordingly in downstream applications.

We have designed and implemented a reference-guided aligner, MOSAIK, that addresses these desired traits while offering outstanding alignment accuracy and competitive alignment speeds. In addition to read alignment, MOSAIK offers a suite of modular tools that address duplicate removal, post-alignment filtering, coverage visualization, multiple sequence alignment creation, and import/export functionality.

2.2. Methods

2.2.1. Processing reference sequences

MOSAIK can handle a practically unlimited amount of reference sequences (4 billion reference sequences); however the maximum aggregate reference length is 4 Gb. Alignments to the human transcriptome using more than 95,000 reference sequences are handled easily.

Our aligner supports the full set of IUPAC ambiguous nucleotide characters during pairwise alignment (Table 2.1 and Figure 2.1). This allows users to use reference sequences that have been masked using confirmed dbSNP[78] calls. The ambiguity codes minimize the alignment bias that might be caused when aligning to the reference reads containing SNPs. The ambiguity codes N and X are always interpreted as mismatches by the scoring algorithm and can be used to hard mask reference sequences.

Table 2.1. MOSAIK has full IUPAC ambiguity code support.

IUPAC Code	Meaning
A	A
C	C
G	G
T	T
M	A/C
R	A/G
W	A/T
S	C/G
Y	C/T
K	G/T
V	A/C/G
H	A/C/T
D	A/G/T
B	C/G /T
N	A/C/G/T
X	(none)

The reference sequences are split up into overlapping contiguous k-mers (hashes) and the positions of each is stored in a hash map data structure. The hash map works well for genomes that are smaller than 1 Gb. For mammalian-sized genomes, the hashes are stored in a sparse, direct-access data structure that contains the offset for the associated reference sequence positions in another tightly packed

data structure. Since the hashes are stored in 2-bit notation (allowing for a maximum of four different nucleotides at each position), the most probable nucleotide (A, C, G, or T) is selected when an IUPAC ambiguity code is encountered. The probability for each nucleotide is based on the nucleotide frequencies present in the human genome. This data structure, called the "jump database", features guaranteed O(1) lookups and can be partly or entirely used from disk (explained in detail in Supplementary Figure 1).

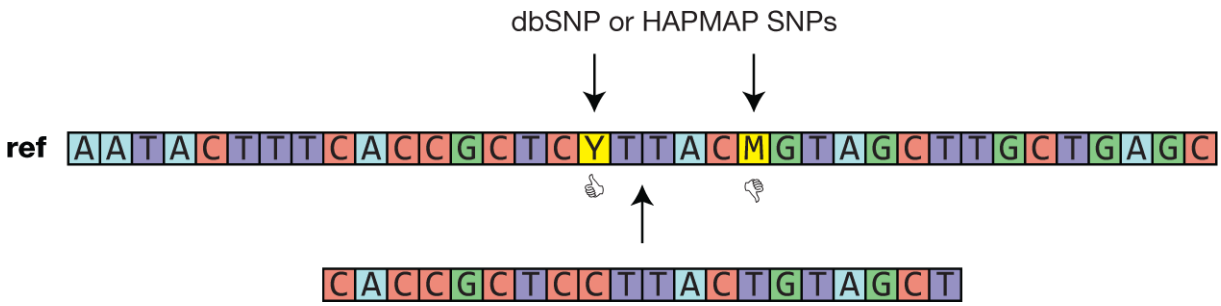

Figure 2.1. Bias against known SNPs can be reduced by using a reference sequence masked with IUPAC ambiguity codes. In the example, the alignment would result in one mismatch (the Y matches the C base, but M doesn't match the T base).

2.2.2. Read alignment

MOSAIK supports various read formats (SRF, FASTA, FASTQ, Bustard, Gerald). Metadata is tracked throughout the alignment pipeline. Attributes such as *median fragment length* and *sequencing technology* directly affect alignment behavior, whereas other attributes, such as *library name*, *sample name*, and the *run identifier*, describe the data set.

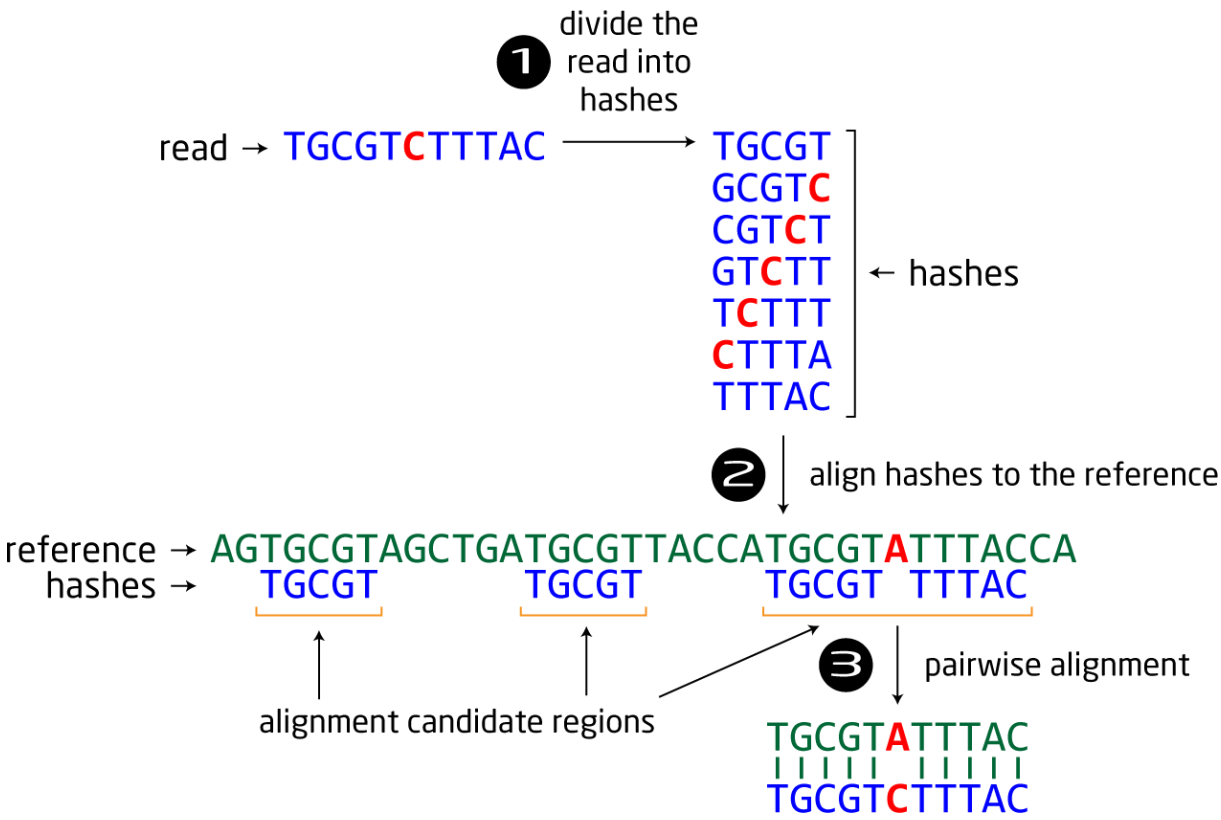

Figure 2.2. MOSAIK alignment algorithm. (1) MOSAIK divides a read into overlapping hashes and then (2) aligns those hashes to the reference sequence. (3) Each cluster of hash alignments (alignment candidate region) is then evaluated with the Smith-Waterman pairwise alignment algorithm. Despite the presence of a SNP in the read (bold cytosine nucleotide), the read is able to be aligned to the proper reference sequence location.

Each read is split into overlapping hashes and the positions for each hash are queried from either the reference sequence hash map or jump database (Figure 2.2). A modified AVL tree[79] is used to cluster nearby hash positions together. The clustering routine is read position dependent and uses a rule set that considers sequencing errors, SNPs, and single-base INDELs. For example, given a hash size of 15 bp and a read of 35 bp, there are 22 overlapping lookups for reference sequence locations. If the read uniquely aligns perfectly to the reference (no sequencing errors, SNPs, or INDELs), all 22 hashes will succeed in finding the proper reference location and the AVL tree will consolidate those hits into one alignment candidate region.

However, if only one hash succeeds in finding the proper reference location because of sequencing errors, an alignment candidate region is still present in the AVL tree. Each alignment candidate region is pairwise aligned using a banded Smith-Waterman-Gotoh alignment algorithm[56,80]. When aligning Roche 454 reads, a modified Smith-Waterman scoring matrix is used to assign a lower gap open penalty when a gap occurs in a homopolymer region. If the alignment was performed in color space (AB SOLiD), it is converted back into base space. Alignments are discarded if the user-defined thresholds, such as the maximum number of mismatches or the minimum alignment length, are not met.

Our read alignment algorithm is heavily multithreaded. Multithreading allows MOSAIK to align reads in parallel using multiple processors, while allowing it to use the same memory footprint as if only one processor was used. One of the difficulties in creating multithreaded programs is that all file and memory operations need to be synchronized. Data corruption may result when one alignment thread writes to a file while another thread is reading from the same file. Scalability is another challenge faced in parallel programming. If a program uses eight processors, one would expect eight times the performance. However, the overhead from synchronizing threads during file operations as well as raw disk and memory throughput can degrade performance if too many processors are being used. Since MOSAIK was designed with scalability in mind, the performance degrades only 17 % when using 24 processors on the same machine (Figure 2.3).

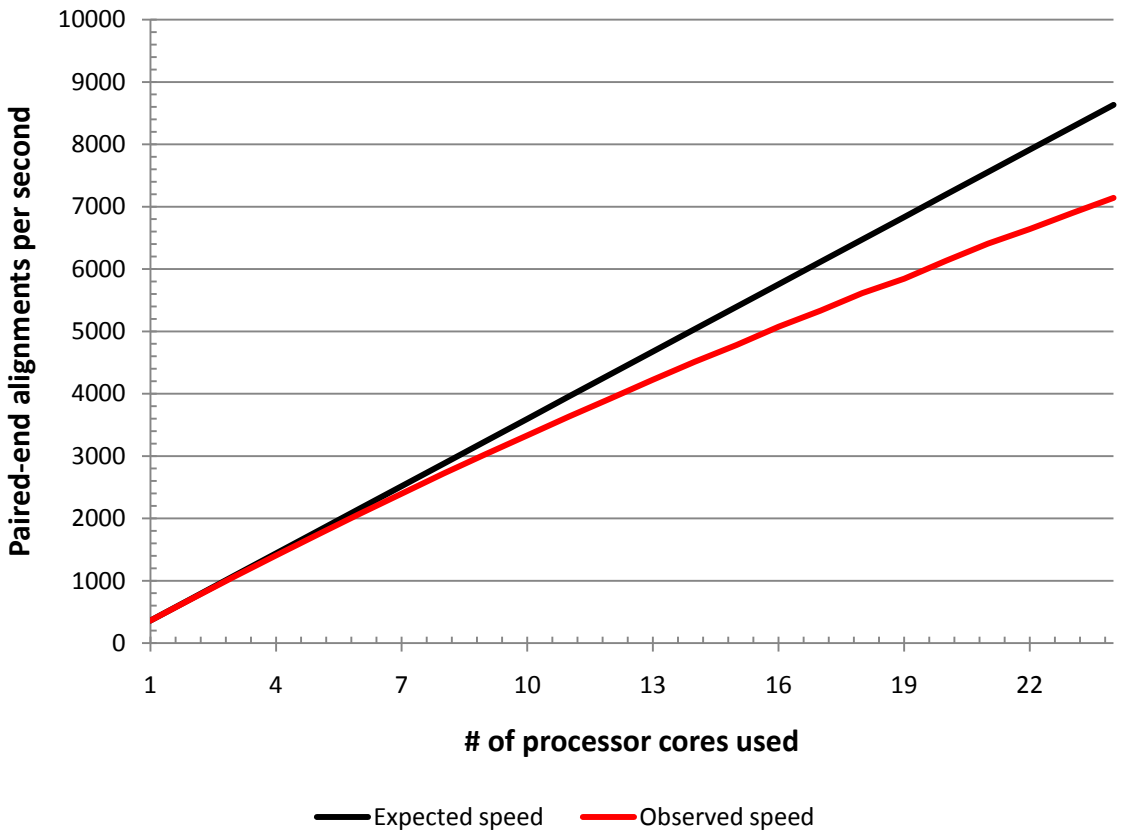

Figure 2.3. Performance improvement as more processor cores are used for one MOSAIK instance. Many multithreaded programs scale well initially, but reach a performance plateau when using many threads. When using 24 cores, MOSAIK still maintains an outstanding parallel efficiency of 83 %.

2.2.3. Mate-pair and paired-end rescue

Each mate sequence in a mate-pair or paired-end read is aligned individually. Inefficiencies in the clustering algorithm, the repeat structure of the genome, and parameters optimizing alignment speed can sometimes prevent a complementary mate sequence from being aligned. To remedy this situation, a local alignment search algorithm has been implemented which performs a Smith-Waterman alignment in the region proximal to a uniquely aligned mate (Figure 2.4). An alignment is considered to be rescued if it conforms to the user-specified thresholds and exhibits the expected order, orientation, and fragment length. Even if an aligned read already

has two uniquely aligned mates, the local alignment search will be performed on both mates.

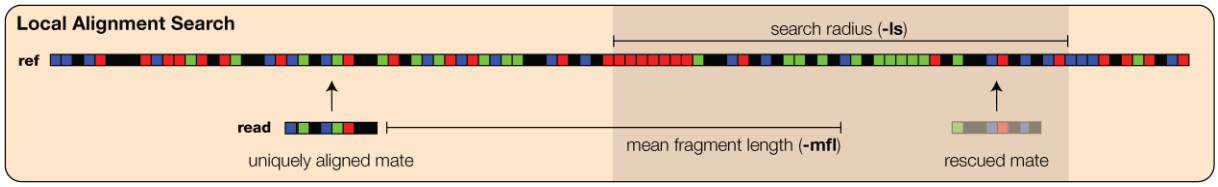

Figure 2.4. When aligning paired-end or mate-pair reads, MOSAIK has the ability to locally search for a missing mate within a user-specified search radius.

The local alignment search works well with paired-end and mate-pair reads that have small fragment lengths (< 500 bp). During normal read alignment, the aligner has information about where candidate regions are located before performing a Smith-Waterman pairwise alignment. In the local alignment search, the candidate regions within the search radius are unknown and therefore the banded Smith-Waterman algorithm cannot be used. Read pairs that have small fragment lengths tend to have tighter search radii, making a full Smith-Waterman approach feasible. With 2.5 kb fragment lengths typical of Roche 454 mate-pairs, the search radius might be 1 kb, which takes much longer to align than a typical Illumina search radius of 150 bp. To remedy this, a virtual dot plot algorithm is in the works that will identify the subregion that enables the banded algorithm to work.

The number of mate sequences that are rescued during local alignment search depends largely on the read length and the alignment parameters chosen for the initial match. As read length increases or more sensitive alignment parameters are

chosen, the aligner is less likely to miss a potential alignment and therefore fewer alignments are rescued (Table 2.2).

Table 2.2. Local alignment search results for Illumina paired-end runs. As the read length increases, the number of rescued mate sequences decrease.

	36 bp	54 bp	76 bp
maximum	5.2 %	4.6 %	0.0 %
average	2.8 % ± 1.1σ	0.9 % ± 1.5σ	0.0 % ± 0σ

2.2.4. Handling Applied Biosystems SOLiD reads

Most downstream applications support base space rather than the color space used by SOLiD reads. Assigning an appropriate value for the maximum number of mismatches is difficult when working with color space aligners. The number of mismatches in color space differs if a SNP has been found or if a sequencing error has occurred (Figure 2.5). A read containing two SNPs and a sequencing error will contain 5 color space bases that differ from the reference sequence. Our approach is to align the reads to a reference that has been converted into color space. However, alignments are converted back to base space immediately after being pairwise aligned. The dibase quality conversion algorithm uses the minimum of the two qualities that overlap a nucleotide in base space. This approach allows the user to specify parameters, such as the maximum number of mismatches, in a manner users expect. Additionally, it enables users to merge aligned SOLiD data sets with data sets from other sequencing technologies.

```
0322122220312220101312112232122230 11130
|||||||||||||||||  |||||||||||
2222031222010131031223212223 0
AATCTGAGAGGCAGAGGTTGCAGTGAGCTGAGATTGTGCC
|||||||||||||||||| |||||||||||
GAGAGGCAGAGGTTGCAATGAGCTGAGATT
```

Figure 2.5. The top alignment depicts a color space alignment (blue) that is equivalent to the base space alignment (green) on the bottom. Both alignments contain a SNP (red), but the dibase encoding results two color space substitutions rather than one base space substitution.

2.2.5. Simulating diploid genomes and short reads

A read simulator was created to evaluate alignment qualities (Figure 2.6). A diploid genome was constructed by adding SNPs and short INDELs to the human reference sequence (NCBI36 / hg18)[16]. Illumina and Roche 454 reads were simulated to study how the differing error models affect alignment sensitivity.

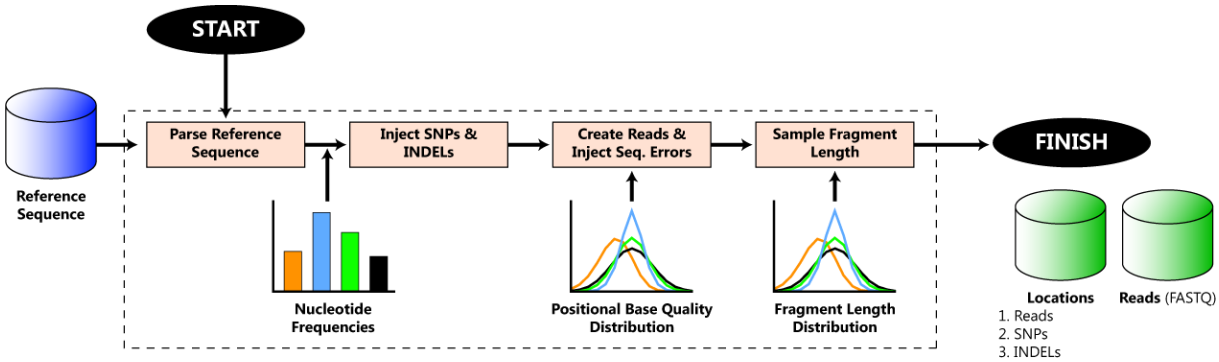

Figure 2.6. Empirical read simulator used to create Roche 454 and Illumina reads. The resulting reads were used to evaluate alignment accuracy and train the single-end alignment qualities.

Empirical base quality, base frequency, and fragment length distributions were collected from 10 Roche 454 FLX runs, 10 Roche 454 Titanium runs, 10 Illumina GA1 runs, and 20 Illumina GA2 runs available in the 1000 Genomes Project. The base quality distributions were used to assign base qualities and to induce substitution, insertion, and deletion errors. The fraction of substitution, insertion, and deletion

errors for each technology was derived from earlier unpublished research by Derek Barnett (Boston College). To study the effects of read length on alignment accuracy, reads were simulated with static read lengths. Various read lengths were simulated for the Illumina platform (36 bp, 51 bp, 76 bp) and the Roche 454 platform (36 bp, 51 bp, 76 bp, 135 bp, 194 bp, 253 bp, 368 bp, 500 bp). For each read length, ten simulated runs were created based on the sampled runs mentioned above. Three million reads were simulated for each Roche 454 run and 10 million reads were simulated for each Illumina run.

2.2.6. Single end alignment quality assessment

Defining the alignment quality model

Quality scores are calculated for each alignment. Similar to base qualities, alignment qualities give the probability that a read has been misaligned. Two alignment quality models were created using the simulated data set: the insertion/deletion model (Roche 454) and the substitution model (Illumina). The Roche 454 model exhibited predominantly insertion and deletion errors, whereas the Illumina model exhibited mostly substitution errors. Initial analysis showed that read length, read complexity, number of mismatched bases, and genome size were strong predictors of alignment accuracy. Read complexity was binned according to 0.1 bit increments of information content (Shannon's entropy[81]). Twenty-one read complexity bins were created ranging from 0.0 to 2.0 bits of information.

One of the problems with using a binning strategy is the goal of filling as many bins as possible to provide a large variation of measured alignment qualities. A linear binning strategy is used with information content. This works well because there is a large natural variation in the alignment information content. This is not the case with mismatch ratios. The mismatch ratio (*mmr*) is defined as the ratio of the sum of mismatched base qualities to the sum of all base qualities in an alignment. Short Illumina reads tend to occupy bins representing a mismatch ratio between 1 % and 10 %. Longer 454 reads tend to occupy bins representing a mismatch ratio between 0 % and 2 %. To solve this problem, an exponential distribution was created that specified twenty-one bins corresponding to mismatch ratios between 0 % and 100 % (Equation 1). The last mismatch ratio bin contains all reads that have a mismatch ratio between 10 % and 100 %.

$$\begin{gathered} bin = [0 \ldots 20] \\ mmr = \frac{\sum BQ_{\text{mismatch}}}{\sum BQ} \\ mmr_{\text{lower}}(bin) = 7.87 \times 10^{-6} \cdot \left(\frac{21 \cdot bin}{19}\right)^{3.06} \qquad (1) \\ mmr_{\text{upper}}(bin) = \begin{cases} 7.87 \times 10^{-6} \cdot \left(\frac{21 \cdot (bin+1)}{19}\right)^{3.06} \\ 1.0 \text{ if } bin = 20 \end{cases} \end{gathered}$$

For each pattern (a unique 4-tuple of predictors), a phred-like[82] alignment quality is calculated using the fraction of misaligned reads. Alignment quality stability was checked by changing the number of misaligned reads by one. If the alignment quality changed considerably (> 1), the pattern was discarded. Empty or discarded patterns were assigned alignment qualities interpolated from the stable

patterns. The resulting set of patterns formed smooth multidimensional surfaces indicating that each of the four features correlated well with alignment quality (Figure 2.7).

Training the neural network

A feed forward backpropagation neural network[83] with 30 hidden neurons was trained to perform a logistic regression of the patterns and the calculated alignment qualities (Figure 2.8 and Figure 2.9). The patterns were randomly partitioned into the training, validation, and test data sets which contained 70 %, 20 %, and 10 % of the patterns, respectively.

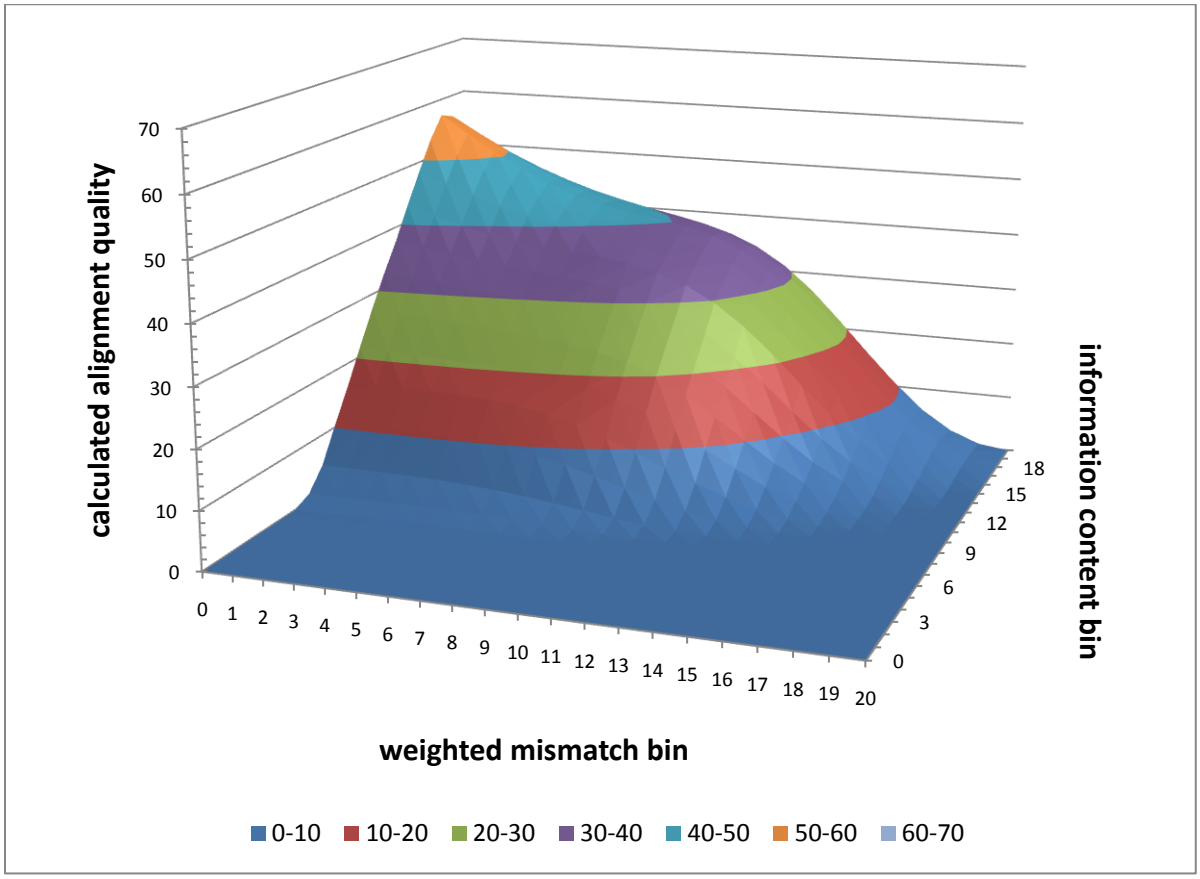

Figure 2.7. The alignment quality landscape for a 36 bp Illumina read being aligned against the full genome. A maximum quality score of 57 occurs when the read has 0 mismatches and 1.8 bits of

information content. The alignment quality increases when the information content bin increases and weighted mismatch bin decreases.

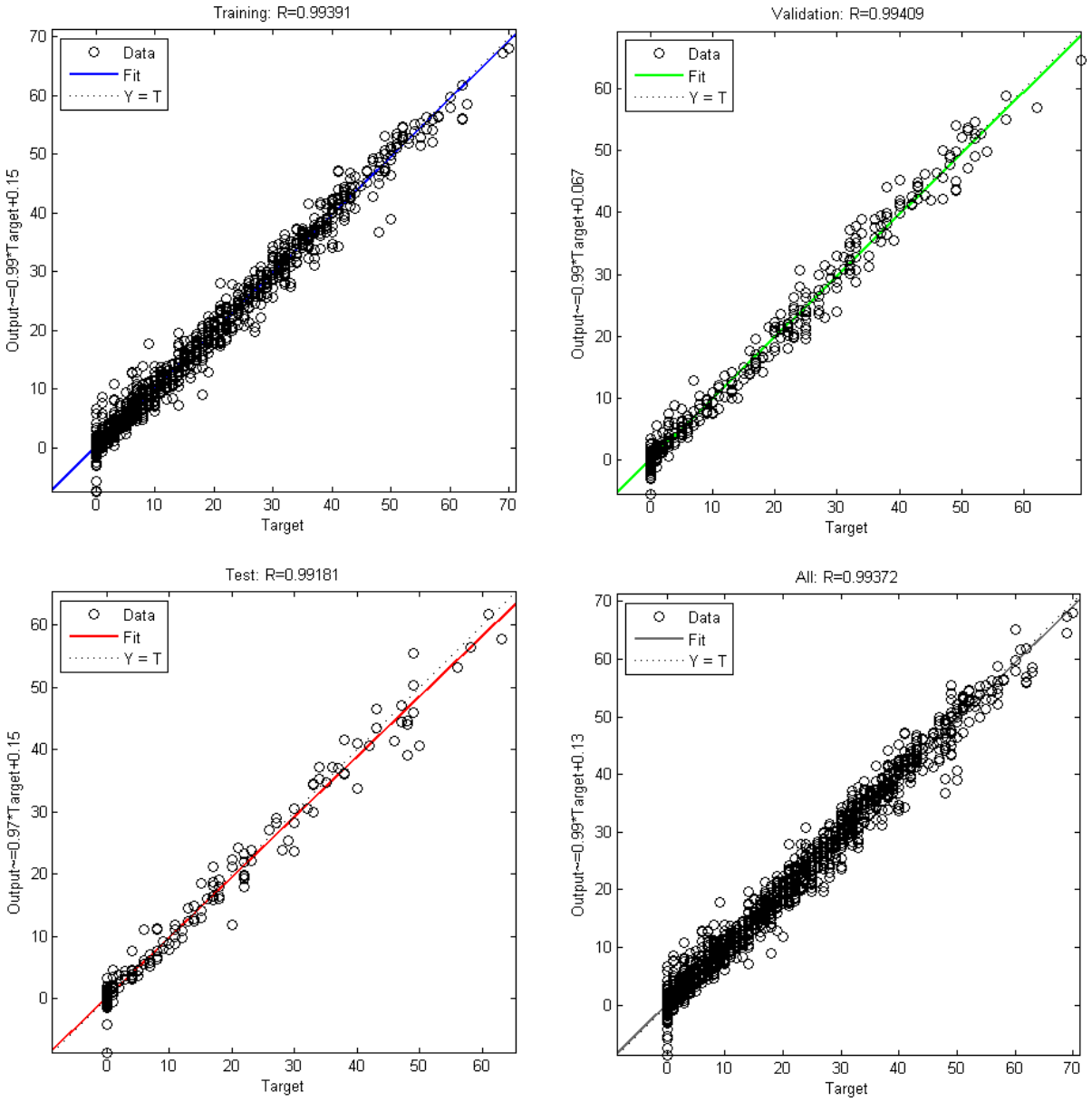

Figure 2.8. These graphs show the correlation coefficient between the measured alignment qualities (target) and the alignment qualities predicted by the neural network (output). Each pattern (black circles) represents a bin with a predefined mismatch ratio, information content, read length, and reference length. For each bin, the number of misaligned alignments and the total number of reads are used to calculate a measured alignment quality. Ideally, the predicted neural network alignment quality (output) should match the measured alignment quality (target) exactly. This would place the patterns on the diagonal. The high correlation coefficients (> 0.99) in all neural network data sets indicate the neural network is working accurately.

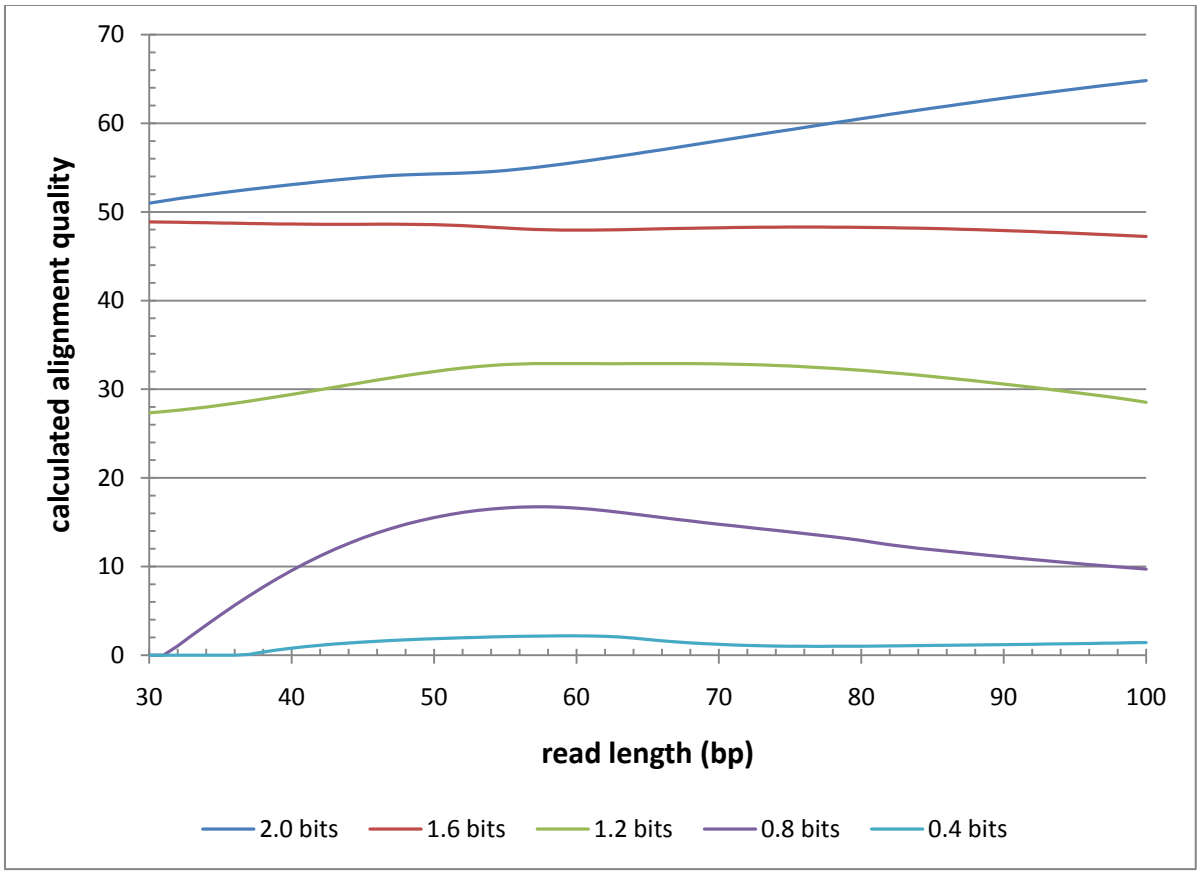

Figure 2.9. Alignment quality sweep for an Illumina read with 0 mismatches being aligned against the full genome. The read length was varied from 30 to 100 bp and checked with various levels of information content (0.4 bits to 2.0 bits) to determine how the machine learning algorithm handled data points not represented in the patterns. The sweep revealed that the learned alignment qualities behaved within expectations when read length and information content was varied. It was interesting to note that increasing the read length had the most impact on alignments with high information content (> 1.6 bits).

Addressing multiply aligned reads

Initially the alignment quality scores were weighted by how many alignment locations were found in the genome. While popular in the field, this method had some unexpected consequences. Any time a read aligned to more than two locations, the alignment quality was effectively reduced to zero. In practice, this meant that non-zero alignment qualities were unique alignments. Our alignment quality definition is "given a pairwise alignment, what is the probability that it was misaligned". Weighting all the alignments using the method described above

muddles the definition by defining "given an aligned read, what is the probability that alignment #2 is misaligned".

The alignment data that is used in our SNP and short-INDEL pipeline is filtered by MosaikSort (discussed in the next section). In practice, multiply aligned reads are removed from single-end alignment archives. With paired-end alignment archives, MosaikSort removes discordant read-pairs. However with concordant read pairs, MosaikSort uses the individual alignment qualities to calculate the paired-end alignment quality (discussed in Section 2.2.8). If the one of the mate sequences in a read-pair aligned to multiple locations, the alignment quality would essentially be reduced to zero if weighting occurred. This restricts our ability to calculate a proper paired-end alignment quality.

Due to these caveats, a choice was made to calculate an alignment quality for each alignment in an aligned read. This method has worked well in both our SNP discovery and structural variation discovery pipelines. The aligner already provides information declaring how many alignments were found for each aligned read. Additionally, features such as reference length and information content are loosely correlated with the likelihood that an alignment has been correctly aligned. For example, a read consisting of 35 adenine bases is more likely to have multiple alignments in the genome than a read consisting of equal numbers of adenine, cytosine, guanine, and thymine bases. Similarly, a read is more likely to align to multiple locations in the genome as the reference length increases. Having used the

human genome when measuring alignment qualities, these issue of weighting for multiply aligned reads has been handled, albeit indirectly.

A weighting method is applied when the user selects parameters that decrease the sensitivity of the aligner (the act and mhp parameters are discussed in Section 2.3.2). It was mentioned in Section 2.2.2 that a 35 bp read has 22 overlapping 15 bp hashes. To increase the alignment speed, the user may choose to only evaluate alignments which have at least three hashes (17 bp) in the alignment candidate cluster and that a maximum of 100 locations will be returned when looking up where each hash occurs in the genome. In this case, the weighting method uses a 17 bp sliding window to calculate the highest ratio of evaluated hash positions to total hash positions. If only 50 % of the hash positions were evaluated in the best interval, the probability that the alignment is correctly aligned is reduced by 50 %.

2.2.7. Filtering aligner output

Downstream applications such as consensus generation and SNP discovery are more sensitive than others to reads that have been misaligned. Single end reads that align to multiple reference locations are filtered to prevent pileups in repetitive regions that often confound SNP callers. The user can override this behavior and allow all single end alignments to pass through.

In a mate-pair or paired-end data set, the first million reads with properly oriented unique mates on each end (uu reads) are sampled to build the fragment length distribution. Fragments shorter than 10 kb are placed into the fragment length

distribution. The minimum and maximum fragment length is derived from the distribution's empirical 99.73 % confidence interval. Based on our tests, the empirical confidence interval is more resilient to outliers than using standard deviations or the median absolute deviation.

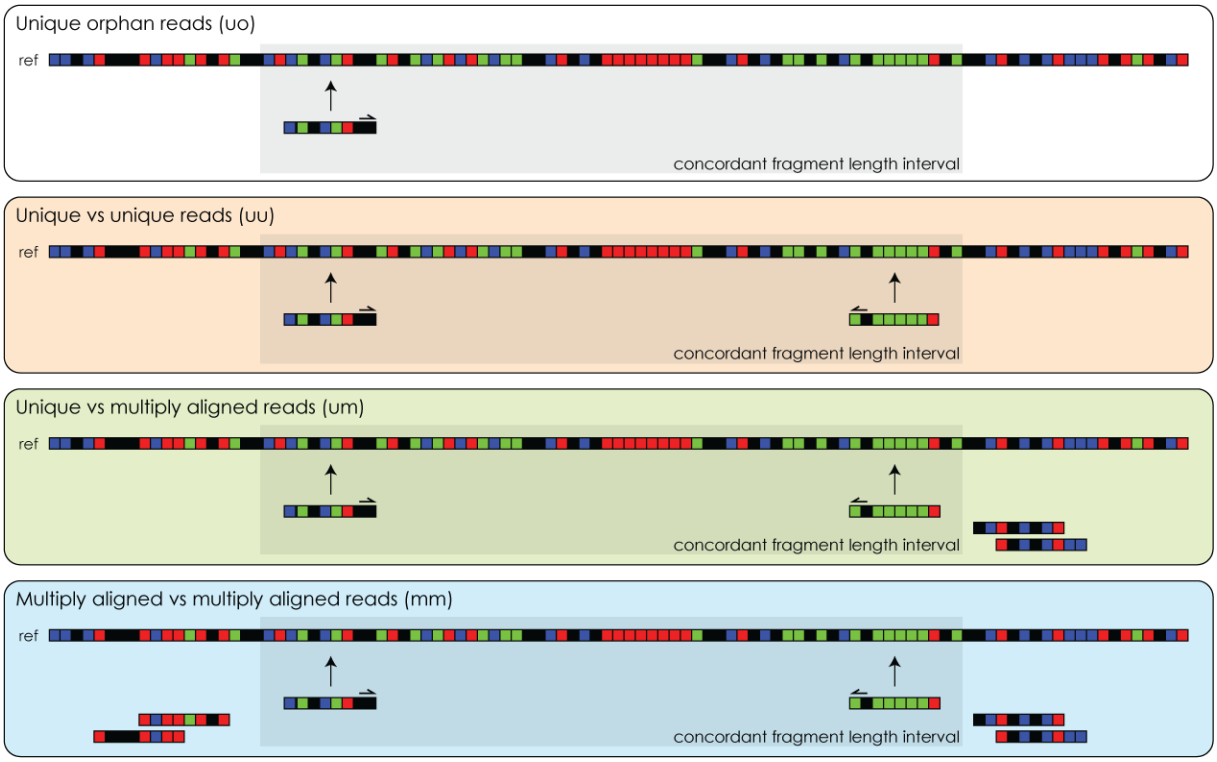

Figure 2.10. Paired-end resolution strategy. The concordant fragment length interval is derived from the 99.73 % confidence interval in the empirical unique vs unique fragment length distribution.

When one mate aligns uniquely to the reference and the other mate fails to align, we consider them unique orphans (uo reads) and allow them to pass through (Figure 2.10). For all reads where both mates align (uu, um, and mm reads), only those mates that have the proper order and orientation are considered. These reads are passed through only if there is a singular combination of mates that occur within the minimum and maximum fragment length. Reads that do not conform to these criteria are discarded. The probability of finding the correct alignment for a read with

multiply aligned mates on both ends (mm reads) is quite low. Normally, this class of reads is skipped, but that behavior can be overridden by the user.

2.2.8. Paired-end alignment quality assessment

Simulated paired-end data sets of varying read lengths were used to assess how the paired-end criteria affect the probability that a read was misaligned. For each paired-end read class (uo, uu, um, and mm), the fraction of misaligned reads for each read length and single end alignment quality were collected and used to calculate the actual paired-end alignment quality (Figure 2.11).

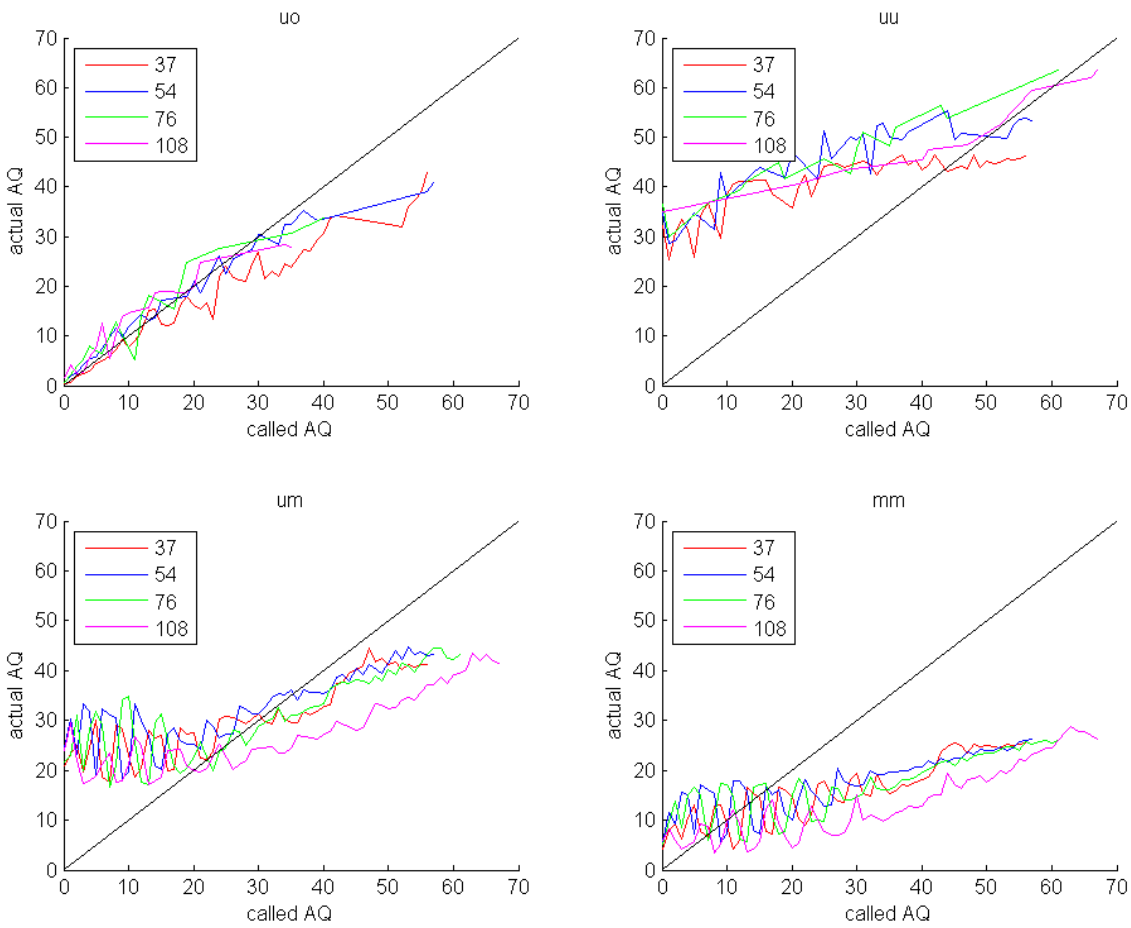

Figure 2.11. For each paired-end read class (uo, uu, um, and mm), the single-end alignment quality (X-axis) was plotted against the actual paired-end alignment quality (Y-axis). Each colored line represents the results for each tested read length between 37 bp to 108 bp.

The paired-end alignment quality equations for each paired-end read class were derived by using multinomial least squares regression (Figure 2.12). Using the read length (l) and the single-end alignment quality score (AQ_{SE}), the regression yielded the following correction equations (Equation 2):

$$AQ_{\mathrm{PE}} = \begin{cases} 0.36l - 0.002l^2 + 1.06AQ_{SE} - 0.007AQ_{SE}^2 - 11.94 \text{ if uo} \\ 0.56l - 0.003l^2 + 0.53AQ_{SE} - 0.003AQ_{SE}^2 + 13.38 \text{ if uu} \\ 0.28l - 0.002l^2 - 0.04AQ_{SE} + 0.006AQ_{SE}^2 + 18.27 \text{ if um} \\ 0.25l - 0.002l^2 + 0.08AQ_{SE} + 0.004AQ_{SE}^2 + 3.84 \text{ if mm} \end{cases} \quad (2)$$

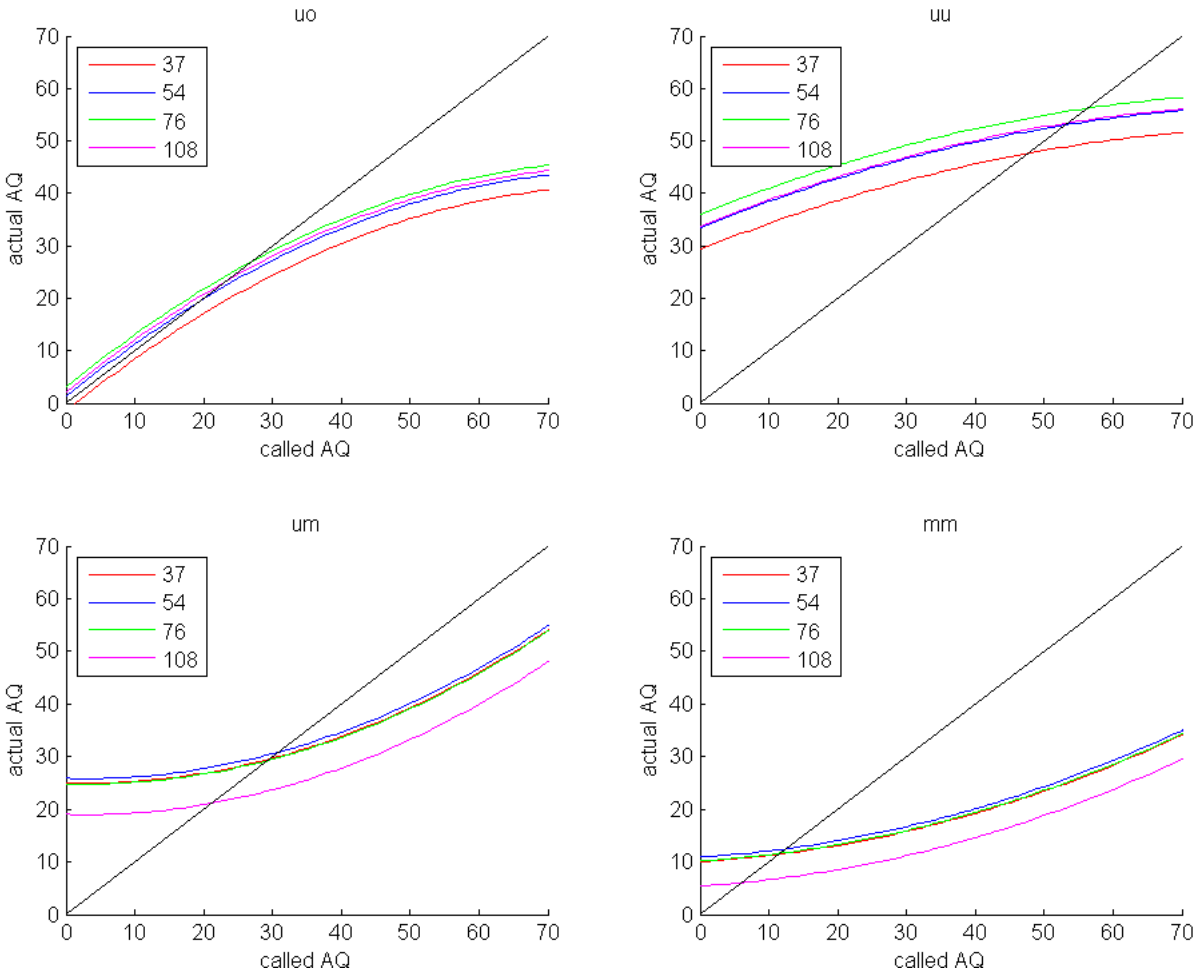

Figure 2.12. Using multinomial least squares regression, second-degree polynomial equations that convert single-end alignment qualities (X-axis) to paired-end alignment qualities (Y-axis) were derived for each paired-end read class. Each colored line represents the results for each tested read length between 37 bp to 108 bp.

The resulting equations are used by MOSAIK to convert single-end alignment qualities to paired-end alignment qualities depending on the paired-end read class.

2.2.9. Sequencing library-aware duplicate filtering

Sequencing library redundancy presents a problem when downstream analyses rely on allele counting. Many nascent duplicate removal tools[58,84] remove duplicate sequence fragments with respect to an entire alignment file. Our duplicate removal tool analyzes a set of alignment files to remove duplicates with respect to the annotated sequencing library. When a set of duplicate alignments are discovered, the alignment identifier with the highest alignment quality is stored in a database. This database is used when filtering the aligner output to discard all duplicate alignments with a lower alignment quality. Paired-end reads are considered duplicates when they share one endpoint but the other endpoint differs by up to 2 bp. Single-end reads are considered duplicates when they share the same start and end coordinates.

2.2.10. Multiple sequence alignment creation

After filtering the aligner output, a multiple sequence alignment can be produced from one or more alignment archives regardless of the sequencing technology. The multiple sequence alignment is created by observing the insertions and deletions in each pairwise alignment. Gaps are then introduced in the reads in order to synchronize the reads with the reference sequence. This algorithm is linear with the number of reads in the multiple sequence alignment.

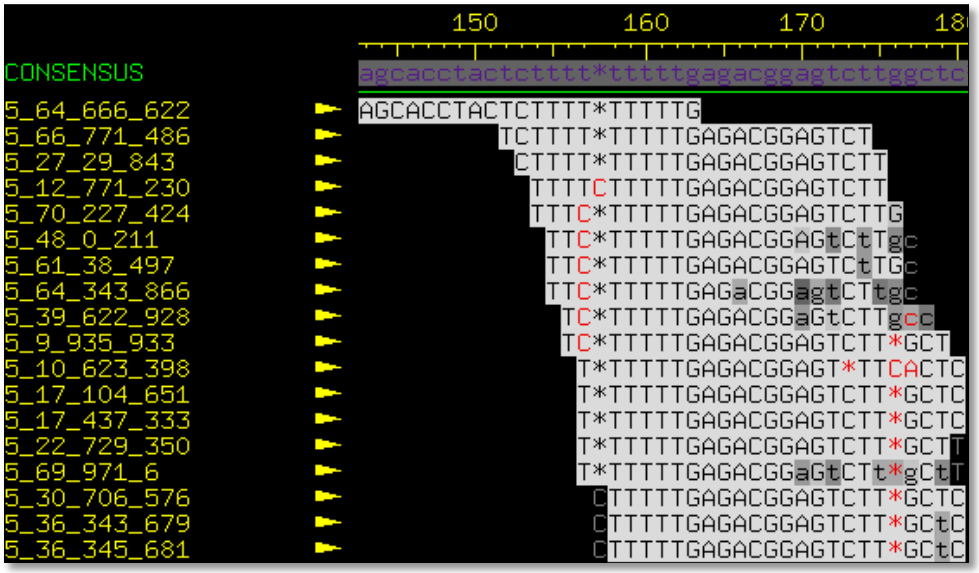

Figure 2.13. Artifacts around heterozygous insertions. In this example, ten reads have a heterozygous insertion of a cytosine nucleotide. However when the insertion occurs close to the end of six reads, they are aligned as substitutions rather than insertions.

Alignment artifacts may result when insertions occur near the ends of reads (Figure 2.13). These artifacts become evident when viewing multiple sequence alignments and might have otherwise been missed when viewing individual pairwise alignments. Using the default scoring scheme, the penalty for opening a gap is higher than the penalty for a substitution. In the example above, the insertion was correctly aligned when it occurred after the fourth base in the read. To reduce the number of INDEL artifacts, an "INDEL cleaner" can be used to adjust the underlying alignments[55]. Currently, the MOSAIK suite does not contain an "INDEL cleaner", but development is expected in the near future.

The multiple sequence alignments can be saved in either the CONSED ace[85] or GigaBayes gig[86] file formats.

2.3. Results

2.3.1. Implementation

MOSAIK is implemented in C++ as a modular suite of programs that are available under an open-source license (GNU General Public License[87]) from our Google Code web site: *http://code.google.com/p/mosaik-aligner/*. Internally, our programs make heavy use of indexed, binary file formats that have been compressed with a real-time compression library, FastLZ[88] (Figure 2.14). Our tools are designed to accept multiple read formats and produce file formats (ace, gig, and SAM/BAM) that are understood by downstream applications (Figure 2.15).

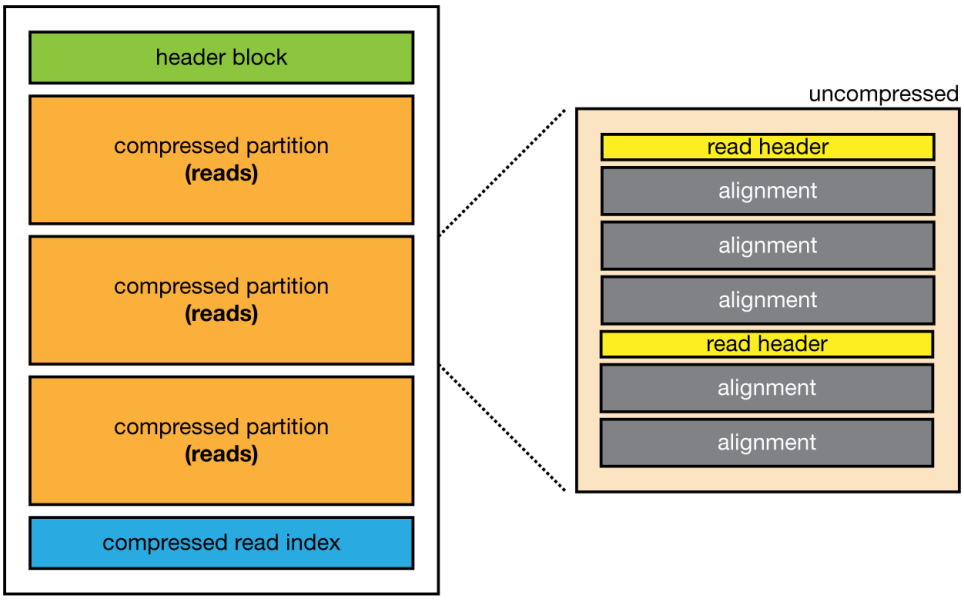

Figure 2.14. Internal organization of the MOSAIK alignment format. For a more detailed view, see Supplementary figures 2 and 3.

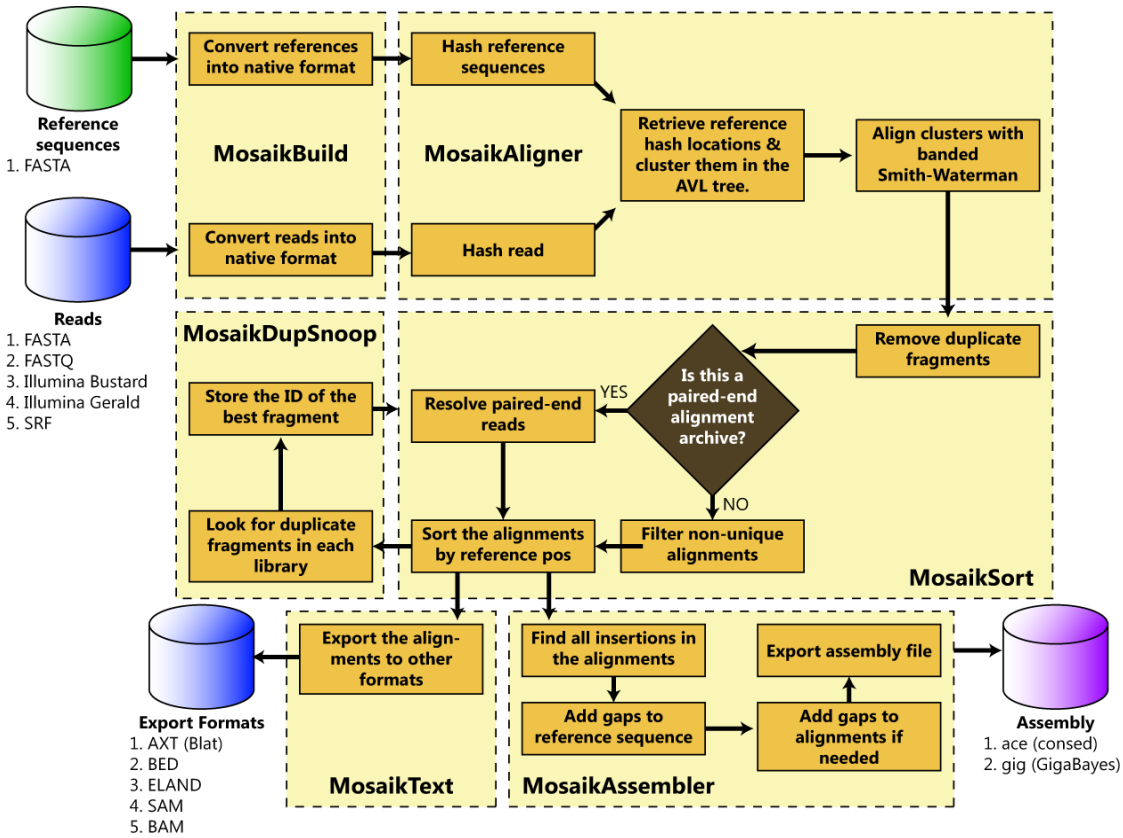

Figure 2.15. Schematic showing major processes within each of the MOSAIK programs and how the key MOSAIK tools relate to one another.

2.3.2. Improving alignment speed

In general, using a larger hash size equates to a faster alignment speed. However, this is often at the expense of sensitivity. For example, using a hash size of 30 on 35 bp reads will align quickly, but reads containing internal sequencing errors or SNPs will not be seeded. In contrast, a hash size of 11 would guarantee that all reads with up to two mismatches are seeded, but performance would suffer. When aligning mammalian reads, a hash size of 15 provides a good compromise between speed and sensitivity.

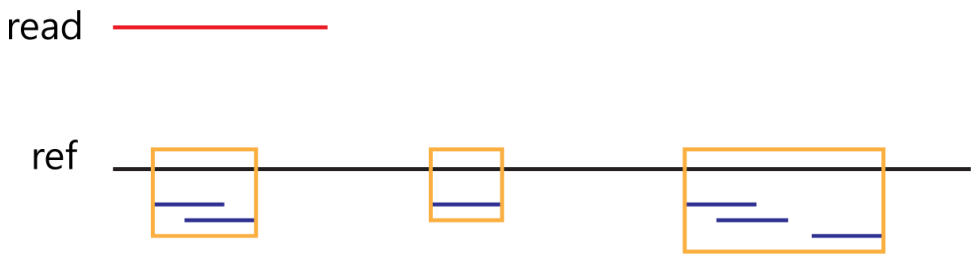

Figure 2.16. Before performing a pairwise alignment, MOSAIK hashes up each read and retrieves the hash positions for each seed in the reference sequence.

In this figure the hash positions are depicted as the horizontal blue lines. When clustering these hash positions, three clusters are formed (orange boxes). Each cluster represents an alignment candidate that will be pairwise aligned.

Perhaps the middle cluster represents a spurious hit due to the repetitive nature of the reference sequence. To prevent spurious hits from unnecessarily using up processing cycles, each cluster can be forced to have a minimum length (the alignment candidate threshold) before being pairwise aligned.

A feature that dramatically improves alignment speed with little impact on accuracy is the **a**lignment **c**andidate **t**hreshold (-act) (Figure 2.16 and Figure 2.17). Normally all clusters are submitted for Smith-Waterman alignment. Initially we imposed a **d**ouble-**h**it criterion (-dh) that two consecutive hashes had to be clustered before being aligned. This double-hit mechanism ensured that fewer spurious hash hits in the reference sequence would cause a full alignment to be performed.

The alignment candidate threshold extends the thought behind the double-hit parameter. An alignment candidate is simply the set of all seeds that form a cluster. The alignment candidate size is the length from the first base in the cluster to the last base in the cluster. For example, if a hash size of 11 is used and two hashes form clusters separated by a SNP, the alignment candidate size is 23. If the act parameter had been set to -act 20, this read would be submitted for Smith-Waterman alignment.

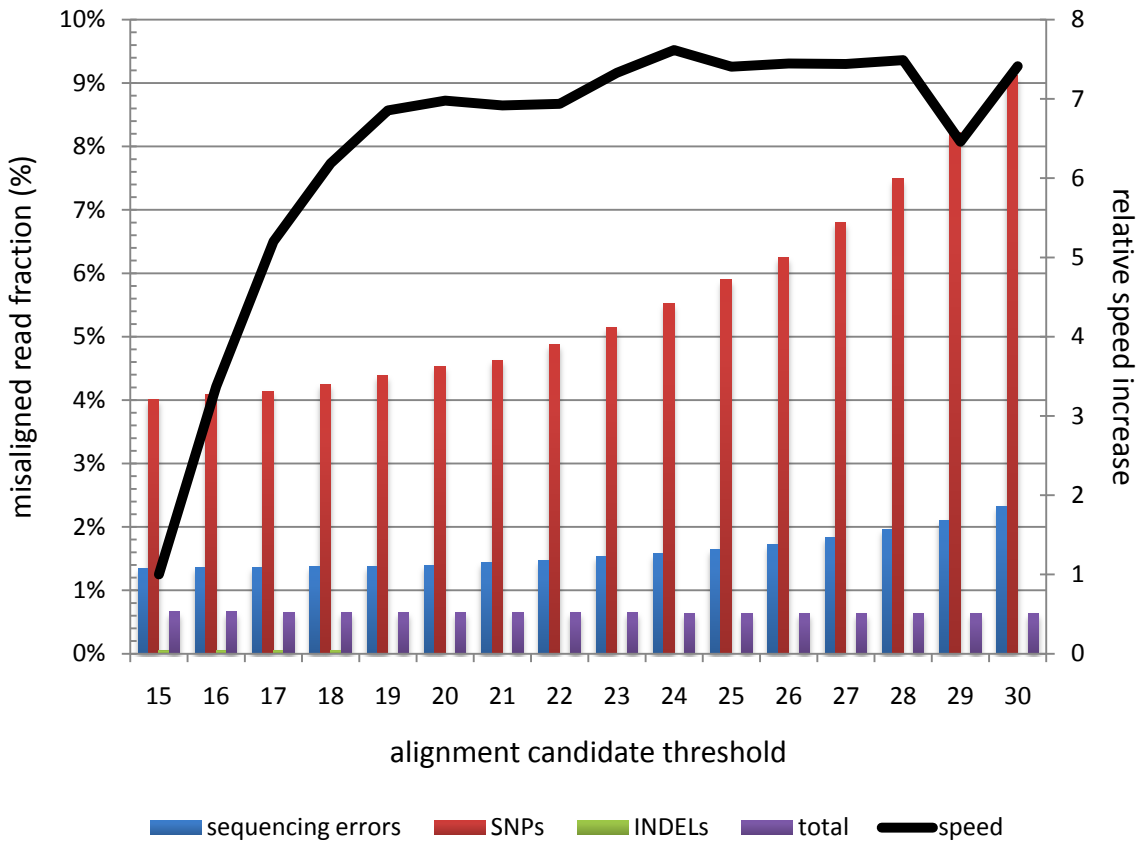

Figure 2.17. Early results showing the effect of the alignment candidate threshold (act) parameter on alignment accuracy. As the act parameter increases, the percentage of misaligned reads (left y-axis) containing SNPs (red bars) increases dramatically. The black line indicates the alignment speed (right y-axis) relative to setting the act parameter equal to the hash size (15 bp). Setting act to 19, results in a 6.8-fold increase in alignment speed.

While scaling up to handle alignments to mammalian genomes, we added a feature (-mhp) that places a **m**aximum number of **h**ash **p**ositions per seed. Alignment representation bias is minimized by selecting a random subset of the hash positions. When using a hash size of 15, each seed has an average of 5.25 hash positions in the human genome (Table 1.2). Limiting the number of hash positions to 100 increases alignment speed significantly, while having little impact on alignment accuracy (Figure 2.18).

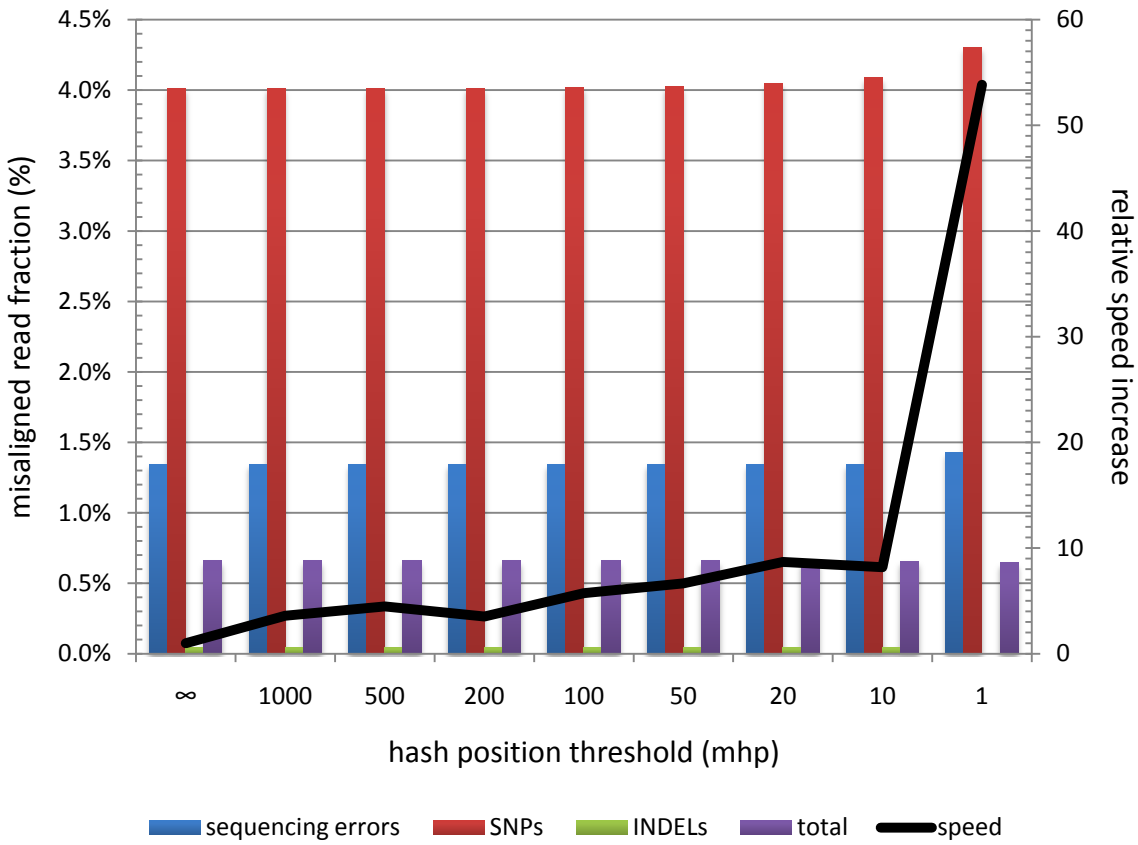

Figure 2.18. Early results showing the effect of the maximum number of hash positions (mhp) parameter on alignment accuracy. As the mhp parameter decreases, the percentage of misaligned reads (left y-axis) increases slightly. The black line indicates the alignment speed (right y-axis) relative to setting the mhp parameter equal to the infinity. Setting mhp to 100, results in a 5.7-fold increase in alignment speed.

2.3.3. Alignment accuracy

Using simulated Illumina data sets generated by James Long at the Translational Genetics Research Institute (TGEN), we evaluated the sensitivity of our alignment algorithms with respect to single-end and paired-end reads spanning from 37 bp to 108 bp (Table 2.3). The number of mismatches was chosen based on previous experience with Illumina data sets of a given read length. More mismatches are typically allowed with longer reads to match the error profile typically seen from Illumina GAII data sets.

Table 2.3. Alignment accuracy on simulated Illumina reads of various lengths. The expected behavior is that both single-end and paired-end accuracy increases as the read length increases. The accuracy of the 76 bp and 108 bp decreases due to the larger number of mismatches allowed.

	37 bp	54 bp	76 bp	108 bp
# of mismatches	4 bp (11 %)	6 bp (11 %)	12 bp (16 %)	15 bp (14 %)
Single-end accuracy	98.67 %	99.37 %	98.46 %	99.24 %
Paired-end accuracy	99.95 %	99.98 %	99.85 %	99.90 %

In the simulated Illumina data sets, MOSAIK had a mean single-end alignment accuracy of 98.94 % ± 0.44σ and a mean paired-end alignment accuracy of 99.92 % ± 0.06σ. If the alignment qualities are properly calibrated, the user can choose a subset of alignments that have higher accuracy (Figure 2.19 and Figure 2.20).

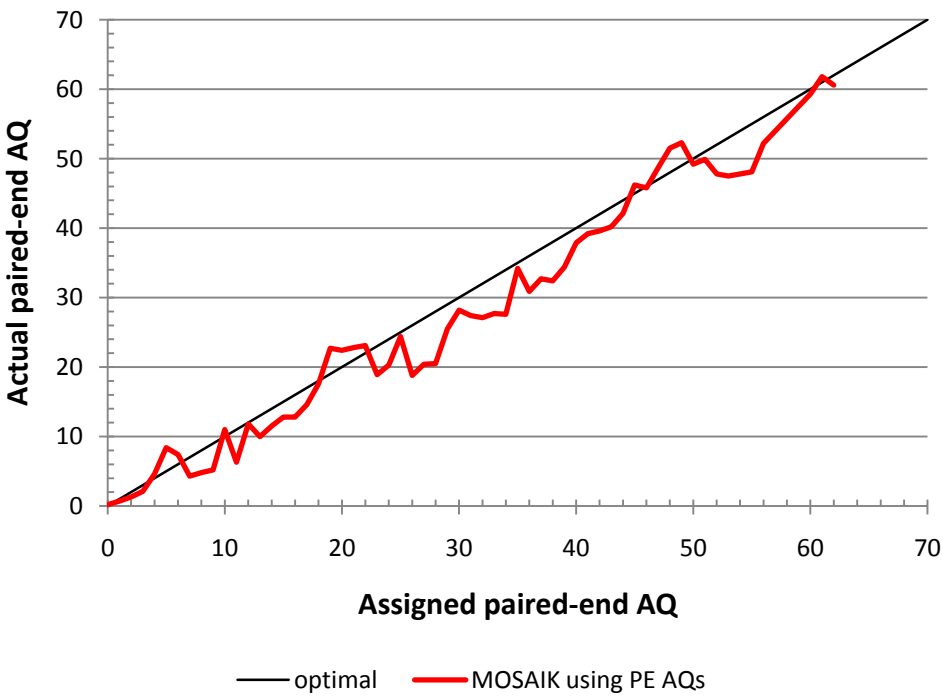

Figure 2.19. Comparing assigned paired-end alignment qualities to actual paired-end alignment qualities. If the assigned paired-end alignment qualities are calibrated well, they will occur close to the diagonal line. The coefficient of determination (R^2) is 0.97 spanning six logarithms of measurement.

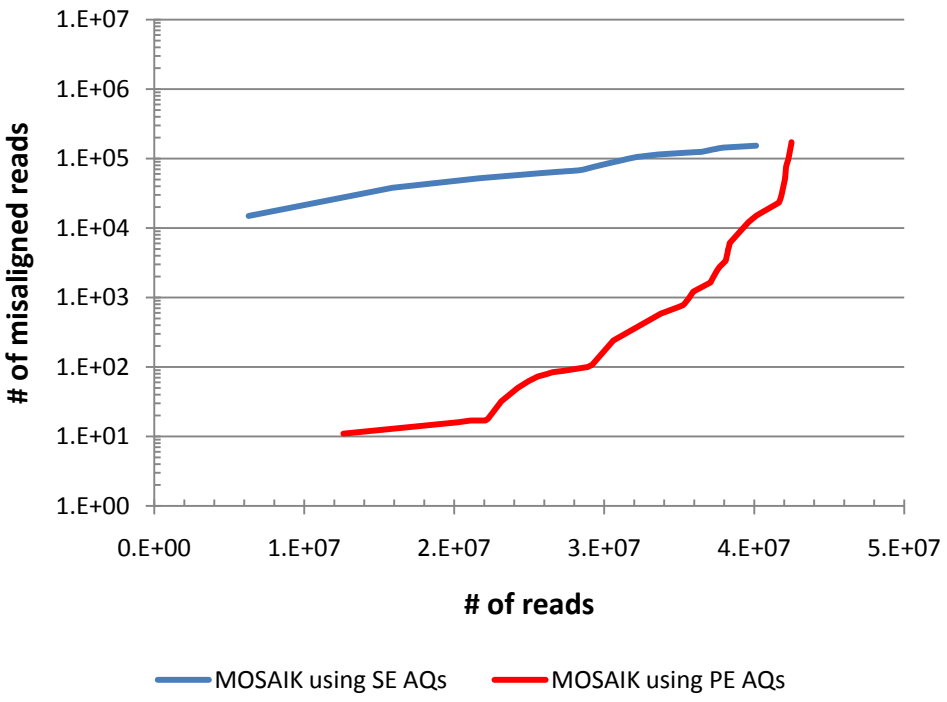

Figure 2.20. The receiver operating characteristic (ROC) curve improved dramatically when the alignment qualities were adjusted for each read-pair class (red curve). Before the calibration, single-end alignment qualities were used for paired-end alignments (blue curve). ROC curves closer to the bottom-right corner indicate that a higher number of reads align with fewer misalignments.

2.3.4. Comparison to other Illumina aligners

As an independent test of how MOSAIK compares with other Illumina aligners, the results of a recent Illumina aligner comparison by Sendu Bala at the Wellcome Trust Sanger Institute are presented. Using the simulated data sets generated by James Long, the accuracy and speed of MOSAIK and the following aligners were compared: Novoalign[89], BFAST[47], srprism (unpublished), Illumina ELAND2[90], BWA[50], SOAP2[51], Bowtie[52], and KARMA[91]. MOSAIK was in the middle of the pack with regards to alignment speed, but was within a factor of two of the fastest aligner, KARMA (Figure 2.21). Of the faster aligners, BWA was the only other aligner that provided gapped alignments in the comparison.

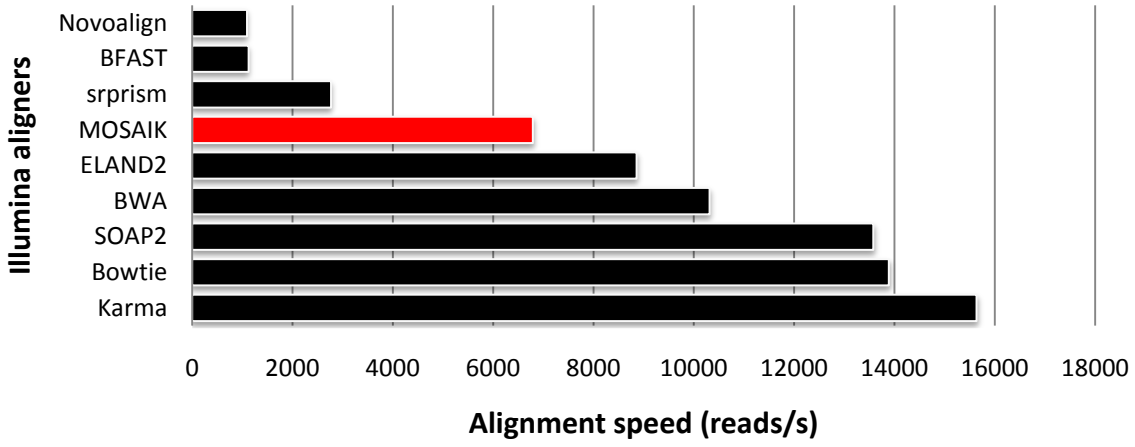

Figure 2.21. Illumina alignment speed. MOSAIK was the second fastest gapped aligner in the comparison study.

Alignment accuracy was evaluated on data sets that mimicked typical data sets used during SNP and short-INDEL discovery. Sendu Bala defined alignment accuracy as the fraction of reads that aligned correctly to the total number reads. Most reads were identical with the reference, but reads with up to five SNPs and reads with INDELs up to 30 bp long were also included. MOSAIK exhibited the highest accuracy of the aligners used in the comparison (Figure 2.22).

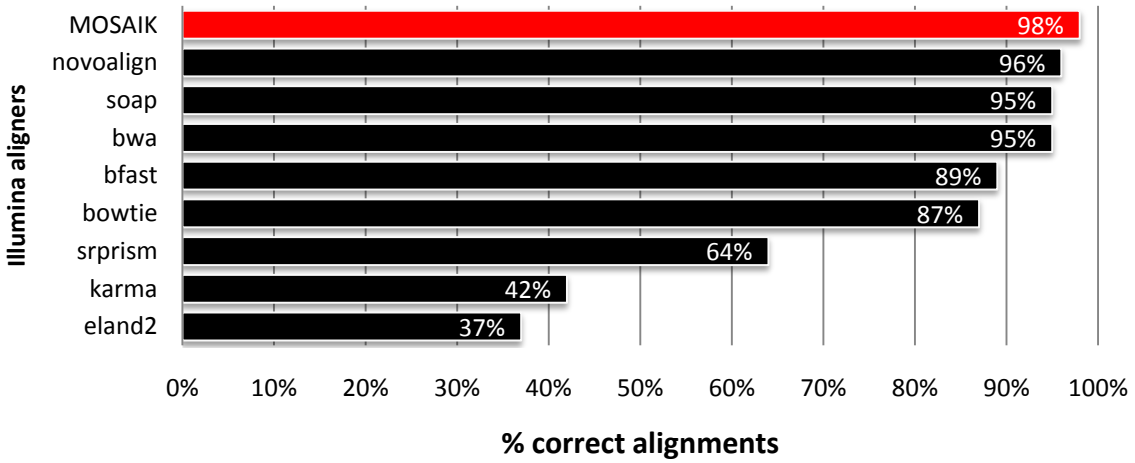

Figure 2.22. Illumina alignment accuracy on the SNP and short-INDEL data set. MOSAIK exhibited the highest alignment accuracy in the comparison study.

In addition to the SNP and short-INDEL data set, a structural variation data set was created. This data set contains reads that are identical with the reference, reads with INDELs up to 30 bp, reads with 200 bp INDELs, and reads with large 20 kb inserts. Unfortunately, the test was run with our post-alignment filtering program that removes discordant read-pairs from a data set. Despite the flawed methodology, MOSAIK was in the middle of the pack by aligning 75 % of the alignments correctly. At the time of the comparison, MOSAIK was not been optimized for aligning reads that are interrupted by 200 bp INDELs. However since the comparison, the local alignment search routines have been adjusted to successfully detect most of the 200 bp deletions in the data set.

2.4. Summary

Since MOSAIK was released as an open source project last fall, a healthy user community has been steadily growing. In the first 24 hours, MOSAIK binaries were downloaded 207 times and the full source code was downloaded 35 times from *http://code.google.com/p/mosaik-aligner/*. Users indicate that MOSAIK is easy to use, highly configurable, and contains handy utilities that give support to their analysis projects. The users show their interest by requesting new features and a select few even proactively provide patches to fix bugs that have been reported by other users.

There are several aspects of MOSAIK that need improvement. MOSAIK currently uses approximately 20 GB of RAM when aligning reads to the human genome. To reduce the memory footprint, alternative data structures and algorithms

are being explored. By simply aligning reads to one chromosome at a time, the memory footprint can be reduced to around 4 – 5 GB of RAM. The use of compressed suffix arrays by the Burrows-Wheeler transform[92], as well as Burkhard-Keller trees[93], are being considered as well. MOSAIK has robust support for both Illumina and Roche 454 sequencing technologies. Currently, MOSAIK development is also bringing Applied Biosystems SOLiD and Helicos support up to the same level.

The comparison study highlights an aspect that has undergone heavy development during the last couple of years - accuracy. The primary focus of our lab, genetic variant discovery, demands that all upstream tools produce the most accurate results possible. Within the 1000 Genomes Project (discussed in Section 3.3), our lab has produced the best SNP, INDEL, and structural variation calls at various stages. One can speculate that a small part of that success is because of the underlying aligner.

3. Re-sequencing applications enabled by MOSAIK

3.1. Whole-genome sequencing and variant discovery in *C. elegans*

3.1.1. Introduction

In 1998 the decoding of the first animal genome sequence, that of *C. elegans*, was published[94]. *C. elegans* was first suggested as a model organism in the 1960s by Sydney Brenner, and subsequent work produced a physical map of its genome[95]. As a result, the *C. elegans* genome sequencing project formed the cornerstone of efforts ultimately aimed at decoding the human genome[16,17]. The entire *C. elegans* community has benefited enormously from the availability of the genome sequence and the ever-improving genome annotation[96], and from the comparative power of the availability of sequenced genomes for *C. elegans* relatives, such as *C. briggsae*[97].

The emerging availability of massively parallel sequencing instrumentation provides the capability to resequence genomes in a fraction of the time, effort and expense required for the initial assembly. We sequenced an isolate of the *C. elegans* N2 Bristol strain using the Illumina sequencing platform. Our analyses of these sequences included an evaluation of sequence differences between the two isolates. We revealed possible sequencing errors in the *C. elegans* reference genome, and putative variants that had occurred in our passaged N2 Bristol strain.

Massively parallel sequencing can be applied to strain-to-reference comparisons that reveal genome-wide sequence differences; either for evolutionary studies or for discovering genetic variants that may explain phenotypic variation.

Implementing this application requires a new approach that assesses the fraction of a genome to which short read sequences can be uniquely mapped, because they are more susceptible to multiple placements than are longer capillary instrument–derived sequences. Computational identification and markup of these 'microrepeats' is therefore an important precursor to accurate short-read analysis, and must allow for mismatches resulting from sequencing errors or polymorphisms. We aligned Illumina reads from the *C. elegans* strain CB4858 (originally isolated in Pasadena, California, USA)[98] to the microrepeat-masked N2 Bristol reference sequence, and identified SNPs and small INDELs with a modified version of PolyBayes[60].

3.1.2. Impact on MOSAIK development

Handling the four Illumina runs present in the *C. elegans* data set and aligning those runs against the full genome represented a challenge to the MOSAIK alignment pipeline. Previously, MOSAIK had been tested on several smaller data sets: the Sanger capillary reads belonging to the ENCODE project[99], the Illumina BAC (human chromosome 11) data set, and the CAPON gene Illumina data set from the Chakravarti lab. The largest reference sequence used in these test data sets were the 500 kb references used in the ENCODE project. To handle the increase in read data and the longer reference sequence, it was clear that MOSAIK needed optimization.

Implementing binary formats

One of the many goals in software development projects involves using data formats that are not only human readable, but that can also be parsed quickly by software

tools. To achieve this goal, XML (Extensible Markup Language) was created by the World Wide Web Consortium (WC3). XML is responsible for enabling AJAX type web applications such as Gmail and Google Maps and is also used in productivity tools such as Microsoft Office, OpenOffice, and Apple iWork. Within bioinformatics, XML is available in many annotation, gene expression, protein, and analysis tools[100].

During my master's thesis, I developed a high performance SNP discovery tool, Forage[65], to identify SNPs in the EST sequences from several species of poplar trees. The output from our *de novo* EST assembler, Paracel TranscriptAssembler, was stored in compressed XML files as opposed to the ace files produced by Phil Green's phrap assembler[39]. The compact nature of the XML files used by Forage was one of the many reasons it was more than 100 times faster than other SNP callers. As a result, the initial versions of MOSAIK used a compact XML schema to store the alignment data.

While parsing XML data files was faster than parsing other text-based flat files, a binary read and alignment format was implemented to improve the speed even more. The binary formats required more low-level code to parse the files, but they helped increase the alignment speed on our CAPON Illumina test data set from 8,000 reads/s to 15,500 reads/s. The resulting speed was much faster than the speed of the only competing Illumina reference-guided aligner (ELAND[90]) at the time.

Partitioning the output

The *C. elegans* data set was comprised of two 32 bp Illumina runs for the N2 (Bristol) isolate, one 32 bp Illumina run for the Pasadena isolate, and one 30 bp Illumina titration run where half of the lanes were used for the N2 isolate and half for the Pasadena isolate. In total, I aligned roughly 99 million reads to the entire *C. elegans* genome. When MOSAIK produced the final multiple sequence alignments in the ace format used by PolyBayes, the files proved too large for our SNP caller to handle. At that time, the only read alignment visualization tool able to handle large ace files was CONSED[85], but the long loading times for large ace files was impractical. As a quick fix to the problems encountered with PolyBayes and CONSED, a partitioning feature was added to MosaikAssembler. Breaking up the *C. elegans* multiple sequence alignment output into partially overlapping 1 Mb regions enabled both PolyBayes and CONSED to perform well.

Consolidating hash positions with an AVL tree

Before performing pairwise alignment, MOSAIK hashes up the read into overlapping hashes and then queries a data structure about the genomic locations of these hashes. Initially a naïve algorithm was used to cluster the hash locations with those that were found earlier in the read. To speed up the clustering process, a data structure known as an AVL tree was modified to quickly aggregate new hashes into known clusters of hashes. With the naïve algorithm it took 36.25 seconds in our test data set to aggregate 99,000 hashes into alignment candidates. In contrast, it took only 0.031

seconds to aggregate the 99,000 hashes using the modified AVL tree. This is more than three orders of magnitude faster than the naïve algorithm.

Microrepeat discovery

The algorithms used to produce genomic assemblies perform poorly in regions where genome-wide repeats are prevalent. Even though the human genome was declared complete in 2003, there are problematic portions that scientists still have not been able to sequence. As of the latest build of the human genome (hg19), only chromosome 14 has been assembled as one contiguous sequence[17].

To reduce the number of alignment artifacts caused by genome-wide repeats, tools such as RepeatMasker have been developed to mask out problematic regions before alignment[101]. RepeatMasker uses the Smith-Waterman[56] pairwise sequence alignment algorithm to align a library of known repeats from well-studied model organisms to supplied input sequences. All significant pairwise hits are then masked out before read alignment. In addition, RepeatMasker also masks known vector and *E. coli* sequences.

Besides screening for known repeats, RepeatMasker also masks out low-complexity regions using the dust algorithm[102]. Low-complexity regions complicate the clustering of similar sequences in sequence assembly algorithms.

For organisms where genome-wide repeats are poorly understood, RepeatMasker offers less intrinsic value. Several algorithms have been developed that try to bridge this gap and provide autonomous repeat identification. RBR1[103]

takes a library of EST sequences and splits them up into overlapping hashes. After a baseline for the expected number of hashes is determined, all hashes that occur significantly more frequently than the baseline are marked as repeats. None of the nascent autonomous repeat identification tools, however, attempt to discover microrepeats that are within a certain number of mismatches from the original reference (edit distance).

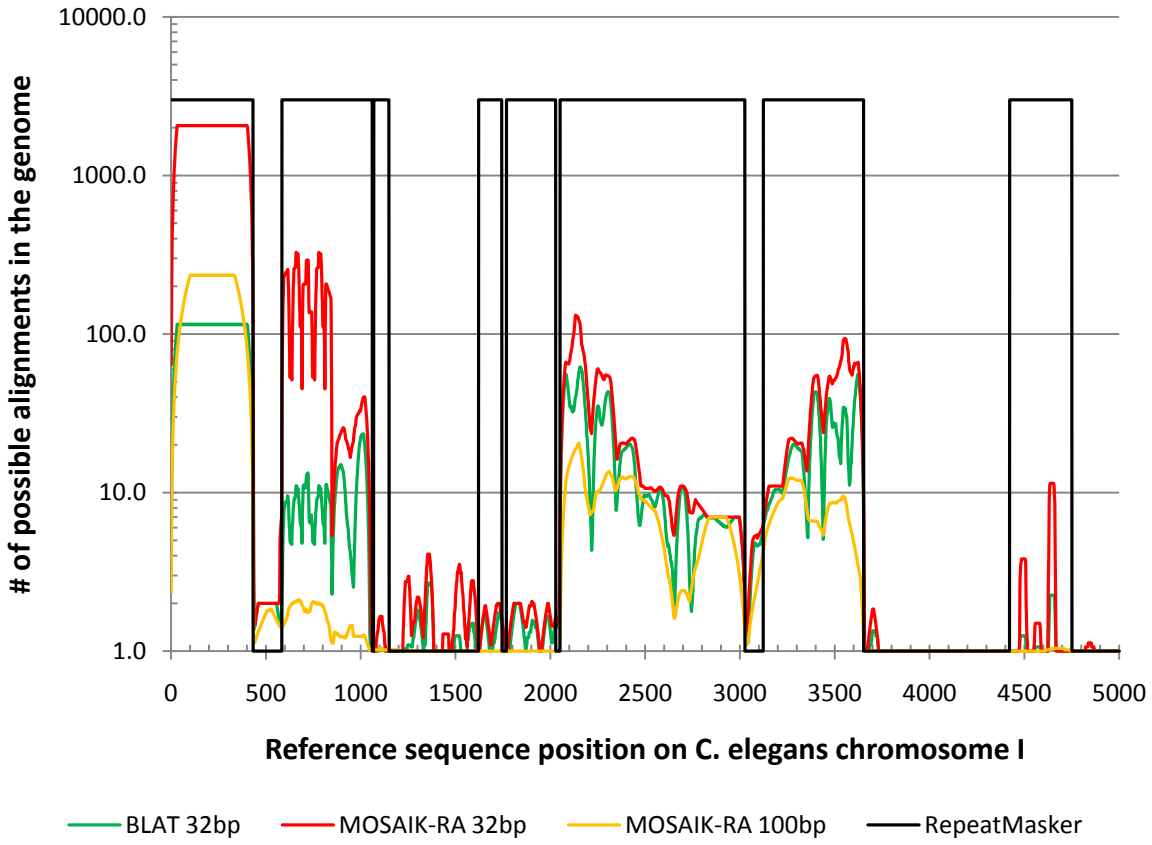

Figure 3.1. Microrepeat discovery in *C. elegans*. BLAT (green), RepeatMasker (black), and MOSAIK-RA (red and orange) were compared when determining repetitive regions in the first 5 kb of the genome. Regions where sequences can be aligned uniquely to the genome are shown at the 1.0 line in the graph. An arbitrary number of alignments (3000) was used to show where RepeatMasker annotations occur. MOSAIK-RA is more sensitive than BLAT at discovering microrepeats. Fewer regions are repetitive to 100 bp reads (allowing seven mismatches) than to 32 bp reads (allowing 2 mismatches).

An experimental version of MOSAIK, MOSAIK-RA, was created to identify microrepeats in the *C. elegans* genome (Figure 3.1). All the overlapping 32 bp sequences were extracted from the genome and aligned back to the genome with an allowance of up to two mismatches. The resulting sequence coverage was normalized by the read length (32 bp). Regions where the normalized coverage was greater than 1.0 were considered microrepeats. These microrepeat regions were then used as a post-alignment filter to screen SNP candidates that may have been called due to alignment artifacts in repetitive regions.

3.1.3. Results

Four full Illumina runs were used in this study: two 32 bp runs for the N2 isolate, one 32 bp run for the Pasadena isolate, and one 30-bp titration run where half of the lanes had the N2 isolate and the other half had the Pasadena isolate. The runs contained 99 million reads, amounting to roughly 24x sequence coverage.

Using the microrepeat detection strategy described above, we noted that roughly 19.8 % of the *C. elegans* genome was considered a microrepeat with respect to 32 bp Illumina reads. This differs from the RepeatMasker results where 14.5 % of the *C. elegans* was considered repetitive (Figure 3.2).

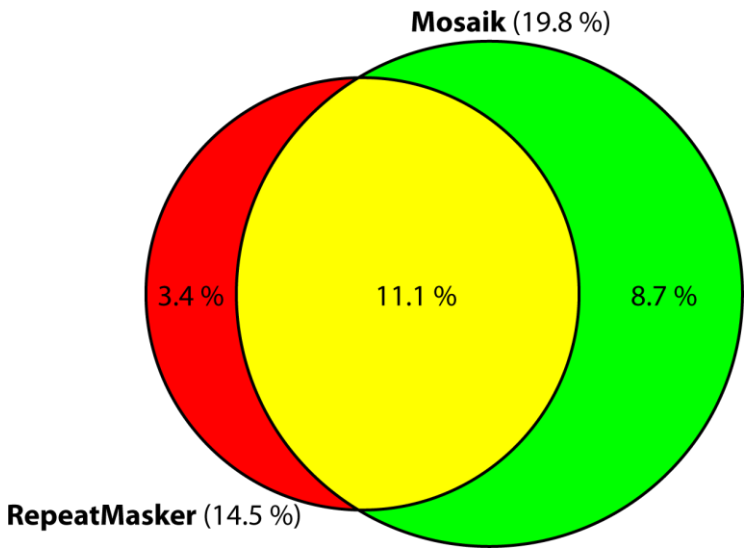

Figure 3.2. The percentage of the *C. elegans* genome that was marked repetitive with respect to microrepeats using MOSAIK vs RepeatMasker repeat annotations.

We ran MOSAIK with a hash size of 16 and allowed up to two mismatches. It took 95 minutes to align the reads (17837 reads/s) and 79 million (79 % of the reads) aligned to either the *C. elegans* or the *E. coli* genomes (Figure 3.3A). Aligning to both genomes allowed us to screen for *E. coli* contamination occurring during sample preparation and from the roundworm gut. It took an additional 100 minutes to create a multiple sequence alignment, partition it into 1 Mb regions, and store it in the ace format (Figure 3.3B).

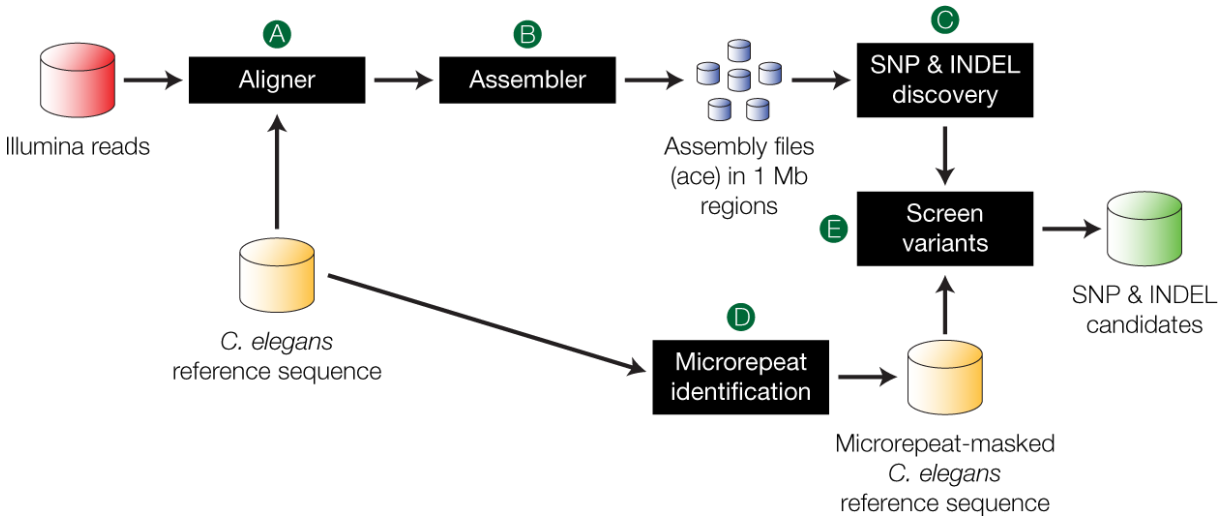

Figure 3.3. *C. elegans* SNP discovery pipeline. (a) MOSAIK was used to align 99 million Illumina reads from the Bristol and Pasadena strains. (b) The resulting 79 million Illumina reads were arranged into multiple sequence alignments and saved as overlapping ace assembly files that represented a megabase region of the genome. (c) PolyBayes was used to call SNPs and short INDELs (d) Using detected microrepeats, a masked *C. elegans* reference sequence was created. (e) SNPs and short INDEL candidates occurring in annotated microrepeats were discarded.

SNPs and short INDELs candidates were identified with PolyBayes by Aaron Quinlan (Boston College) (Figure 3.3C). SNP and INDEL calls with a posterior P(SNP) probability greater than 0.7 were screened using our microrepeat annotations (Figure 3.3E). Roughly 1000 SNPs and INDELs were validated by PCR amplification and Sanger capillary sequencing. The SNP validation rate was 96.3 % and the INDEL validation rate was 93.8 %. Using PolyPhred[104] on the Sanger validation traces, we determined our false negative rate to be about 3.75 %. Validated SNPS and INDELs were assigned WormBase[96] accession numbers pas1 – pas50906.

Our colleagues at the Washington University School of Medicine identified 235 INDELs that occurred in regions where no other Illumina reads could be placed. This suggests an INDEL error rate in the N2 reference genome of about one INDEL in 427 kb. Likewise 544 substitution polymorphisms were identified where more

than one read exhibited the minor allele. This suggests a substitution error rate of about one SNP in 184 kb.

3.1.4. Summary

The *C. elegans* study helped prepare the MOSAIK alignment pipeline for whole genome alignments. The implementation of the binary formats and the AVL tree clustering algorithms improved MOSAIK's alignment speed.

The observation of how microrepeats affect the resequenceability of the reference genome led to ideas on how alignments can be made more accurate despite repetitive regions in the genome.

During this study, an option to partition the multiple sequence alignment output in MosaikAssembler was added. The option relieved downstream applications of the problems of scaling up to large data sets. This mindset of tweaking MOSAIK to benefit downstream analysis has continued throughout the development process and represents one of the many reasons users decide to use our aligner.

3.2. Rapid whole-genome mutational profiling using next-generation sequencing technologies

3.2.1. Introduction

Pichia stipitis is a haploid yeast related to endosymbionts of beetles that degrade rotting wood[105]. It is an important organism for bioenergy production from lignocellulosic materials because of its high capacity to ferment xylose and cellobiose to ethanol[106]. The reference strain was sequenced previously resulting in a completely characterized genome of eight chromosomes totaling 15.4 Mb of sequence[107]. This strain has been subjected to chemical mutagenesis, phenotypic selection, genetic engineering, and adaptive evolution in order to develop strains improved for ethanol production. Chemical mutagenesis and selection resulted in small improvements in ethanol production attributable in part to carbon catabolite derepression. Disruption of CYC1 (cyctochrome c, isoform 1) to create strain Shi21 increased the specific ethanol production rate by 50% and the ethanol yield by 10%; however, the additional mutational events leading to this phenotype were uncharacterized.

Traditional methods for identifying mutations are labor and time-intensive, so we tested the ability of next-generation sequencing technologies to determine the differences in this improved strain's entire genome, relative to the reference strain. We generated high-coverage, whole-genome data sets using single fragment end reads from three next-generation sequencing platforms: Roche 454, Illumina, and Applied Biosystems SOLiD. We assessed these data to determine the effect of

sequence coverage on the accuracy of mutation detection, and to compare the efficiency of the three sequencing platforms for this application.

3.2.2. Results

Three different second-generation sequencing technologies were used to discover SNPs mediated by chemical mutagenesis. Using these data sets, our collaborators at the Department of Energy (DOE) Joint Genome Institute (JGI), found 17 mutations between the parent and mutant strain, where three mutations were caused by errors in the reference sequence. The remaining 14 mutations were validated by Sanger capillary sequencing.

The role of our lab in this study focused on using the read alignment and SNP discovery pipeline to discover the polymorphisms in the Roche 454 and Illumina data sets. The primary focus of this study was to determine the minimum amount of read coverage needed to produce the minimal amount of false positive and false negative SNP calls. The results could then be used by labs considering polymorphism discovery assays using second-generation sequencing technologies to calculate the most economical sequencing strategy.

Two Roche 454 FLX runs and one Illumina 32 bp run were sequenced and analyzed by our alignment and SNP discovery pipeline. Using both runs (11x aligned read coverage), the Roche 454 FLX results had one false positive and zero false negative SNP calls. The false positive in this data set is believed to be caused by an emulsion PCR artifact during library preparation. Subsampling the dataset to use 1.5

runs increased the number of false positives. Using all seven lanes, the 32 bp Illumina run (44x aligned read coverage) exhibited zero false positive and zero false negative SNP calls. Using only three lanes (19x aligned read coverage) yielded the same result with false positives finally occurring when only two lanes were used.

Applied Biosystems performed similar analyses on the SOLiD sequencing platform. Using the Corona Lite data analysis package[108] to map reads and perform SNP discovery, they determined that subsampling their data to 10x coverage yielded zero false positive and zero false negative SNP calls.

3.2.3. Impact on MOSAIK development

Masking nuclear mitochondrial DNA

Numts (nuclear mitochondrial DNA) are copies of mitochondrial DNA that have been transposed into the nuclear DNA[109]. Numts have been detected in many diverse eukaryotic organisms and often manifest as tandem repeats[110,111]. Our collaborators suggested that we mask out all numts in the *P. stipitis* genome to prevent alignment artifacts in those regions.

Using both BLAT and BLAST (with an expectation value of 1E-3), six partial numt regions were identified. A simple tool was created to mask out regions of reference sequences based on annotated locations and was used to mask the numts in the *P. stipitis* genome.

Microrepeat discovery

An experimental version of MOSAIK, MOSAIK-RA, was used to detect microrepeats in the *P. stipitis* genome with respect to each technology and each read length (Table 3.1). The mean read length was used for variable read length technologies (Roche 454 GS20 and FLX) when determining the resequenceability of the genome.

Table 3.1. Percentage of the *P. stipitis* that was masked due to microrepeats.

Sequencing technology	Roche 454 GS20	Roche 454 FLX	Illumina	
Evaluated read length	103 bp	224 bp	27 bp	32 bp
% of genome masked	5.5 %	5.3 %	7.9 %	6.8 %

Read alignment

In addition to the data mentioned in the results section, three additional 26 bp Illumina runs, 533 Sanger capillary validation sequences, and 10 Roche 454 GS20 runs were aligned using MOSAIK. The read alignment parameters were fine-tuned for each sequencing technology (Table 3.2). To aid visualization at the 14 mutant SNP locations, a co-assembly was created in MOSAIK featuring reads from all four of the aforementioned sequencing technologies (Figure 3.4).

Table 3.2. MOSAIK alignment parameters for each sequencing technology. The best hit in the genome was selected for the Sanger validation reads, regardless of the number of mismatches*.

Sequencing technology	Roche 454 GS20	Roche 454 FLX	Illumina	Sanger*
Allowed mismatches	5 %	5 %	2	-
Minimum aligned read length	95 %	95 %	95 %	-
Hash size	21	22	8	10

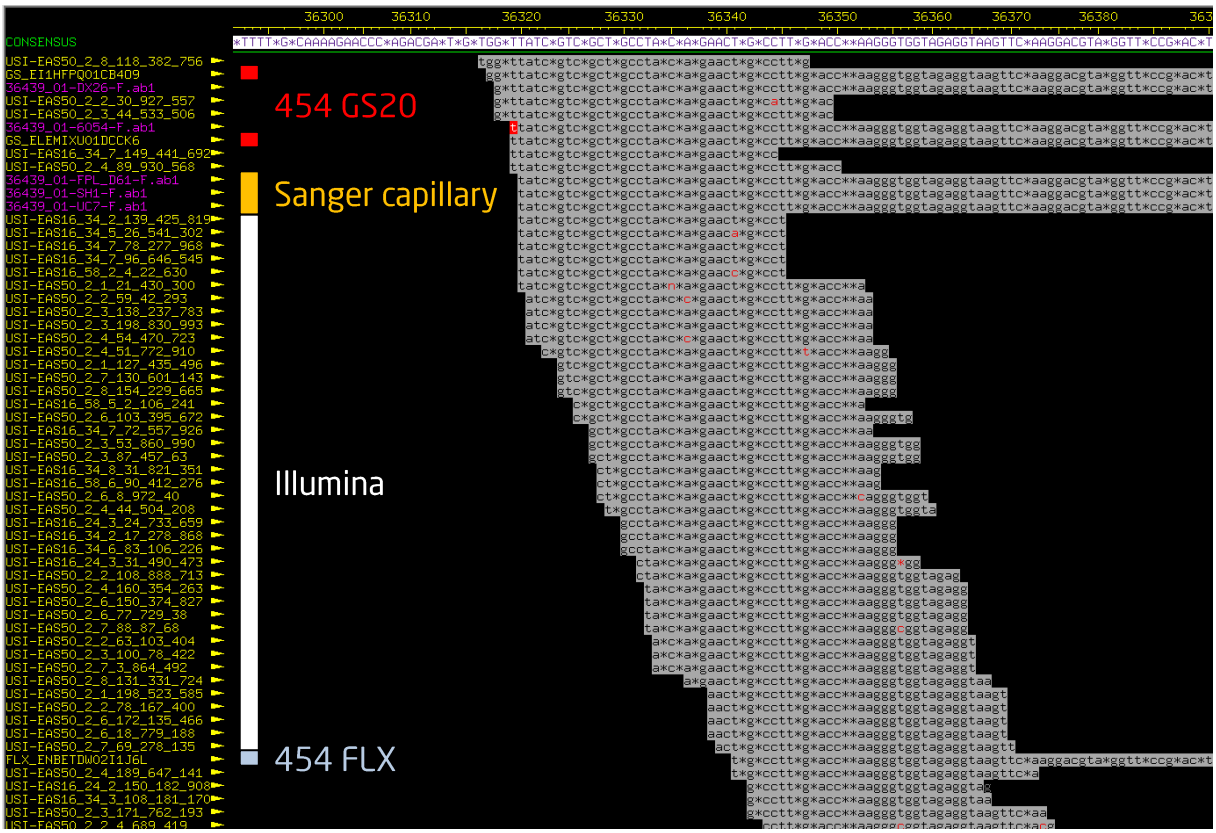

Figure 3.4. MOSAIK Co-assembly. This screen capture of CONSED shows four different sequence technologies represented in the same multiple sequence alignment. Sanger capillary validation sequences are shown in orange, 454 GS20 reads are shown in red, 454 FLX reads are shown in light blue, and Illumina reads are shown in white.

High performance computing

In the period leading up to the *P. stipitis* study, several performance improvements in the MOSAIK code base were being investigated. An experimental version of the aligner was implemented using the Message Passing Interface (MPI)[112]. MPI is a language-independent communications protocol that is used to coordinate parallel computing tasks across many computing nodes. Using MPI, MOSAIK was able to use the entire computing cluster in the Marth lab to coordinate alignment tasks. In an automated fashion, binary read files were distributed and aligned by each worker node. When all the worker nodes were finished, the master node would collect all the alignment statistics, as well as the binary alignment files from each node.

The MPI-enabled MOSAIK worked well for this study but was later abandoned due to many incompatible MPI implementations in the market. If MOSAIK had been released with MPI support, it would have worked on only a few computational clusters because of the incompatible implementations. As a replacement, a platform-independent multithreading class (a programming construct used in object-oriented programming) was implemented that would accomplish the same goal yet worked on all modern operating systems.

Berkeley DB

The main data structure that MOSAIK uses to store the hashes and the associated reference locations is a hash map. Hash maps offer fast lookup performance at the expense of using large amounts of memory. During this study, an alternative data structure with a smaller memory footprint, the Berkeley DB[113], was being explored. Berkeley DB is a disk-based key-value data store that can be organized as a linear hash map, a queue, or a sorted, balanced tree. One of the beneficial aspects of the Berkeley DB is that it provides a cache of the recently used key-value pairs. With MOSAIK this means the most commonly used hashes and associated reference locations are likely to remain in cache memory. By specifying the amount of cache memory used, the user can fine-tune the trade-off between memory usage and alignment speed. Eventually, Berkeley DB support was dropped in favor of the jump database (introduced in Section 2.2.1 and explained in detail in Supplementary figure 1).

3.2.4. Summary

The *P. stipitis* study was our first opportunity to evaluate several different sequencing technologies on the same underlying reference genome. Previously, we had examined the error profiles in the Illumina and Roche 454 sequencing technologies. In this study, however, the effects of the underlying error profiles were finally seen in the context of the minimum coverage needed to detect the mutant SNPs. In general, sequencing technologies that exhibited higher error rates required deeper coverage when detecting mutant SNPs.

This project marked a transition period in the development of MOSAIK. During this study, the reference sequence was carefully masked according to microrepeat structure found at different read lengths. This encouraged the development of more sophisticated alignment algorithms and data structures that would allow MOSAIK to handle repetitive regions without microrepeat masking.

3.3. Genetic variant discovery in a deeply sequenced European trio

3.3.1. Introduction

The 1000 Genomes Project is an international consortium whose primary aim is to construct a deep catalog of human genetic variant at minor allele frequencies of 1 % or higher[114]. Several strategies are being tested during the pilot phase of the project. These include evaluating many samples at low coverage, evaluating two sets of trios at high coverage, and evaluating targeted sequencing of exons and other functional elements. Our study involves read alignment and genetic variant discovery in the high coverage trio from a Utah family having European ancestry (1000 Genomes Project pilot 2)[114].

3.3.2. Pre-analysis development

Read name nomenclature

For GigaBayes to decode the origin of each read and the relationships among all individuals in the trio, a prefix was prepended to each read before alignment. The following colon-delimited format was used: "**NA12878:1463:NA12892:NA12891:F_**". NA12878 indicates the Coriell Institute for Medical Research identifier[115] for the current sample with the family identifier 1463. NA12892 is the sample's mother and NA12891 is the sample's father. The final field denotes that the gender of the sample is female.

Paired-end support

The 1000 Genomes Project was the first opportunity our lab had at aligning large data sets and handling paired-end read data. Initially MOSAIK was designed to align the fastq files for each paired-end mate separately. Two separate alignment files were produced that had to be sorted and combined before searching for concordant read-pairs. This strategy did not scale well with larger data sets.

The solution was to update the binary file formats used throughout the MOSAIK pipeline to store read-pair information in the same read record. MosaikBuild (Figure 2.15) was modified to parse up to two read files simultaneously and MosaikAligner (Figure 2.15) was updated to store the alignments from the two read-pairs in the same read record. This obviated the need for preprocessing before evaluating read-pair concordance and it reduced the processing overhead for the structural variation analyses being performed in the lab.

Post-alignment filtering and sorting

MosaikSort was created to filter and sort the alignments according to the reference position. For single-end reads, non-unique alignments are filtered. For paired-end reads, discordant read-pairs are filtered. These filters reduce the probability that misalignments will complicate variant calling. The sorting enables MosaikAssembler (Figure 2.15) to produce a multiple sequence alignment several orders of magnitude faster than previous versions.

Native support for GigaBayes

Instead of using the ace file format that CONSED uses for visualizing multiple sequence alignments, GigaBayes uses a native, binary file format (.gig files). Normally ace files are created in MosaikAssembler and GigaBayes converts them to gig files in GigaBuild. The entire process was painfully slow. To speed up the variation calling pipeline, an optimized gig file writer was added to MosaikAssembler.

SNP evaluation program

GigaBayes produces a detailed output file, but does not include any utilities to assist in filtering and analyzing the data. To remedy this, a SNP evaluation program was created that filtered GigaBayes results according to inter-SNP distance and P(SNP). Also, it computed the overlaps between our called variants and HapMap3 genotypes, confirmed SNPs from dbSNP, and confirmed variant calls from other labs (Supplementary Figure 4).

3.3.3. Initial alignment and SNP calling

From November to December 2008, the first iteration of alignments and variant calling was performed. This gave us the opportunity to streamline and fix bugs in our pipeline.

Duplicate filtering

In previous projects, the sequencing libraries produced by collaborators exhibited high complexity. A sequencing library is said to have low complexity when the same

fragment is sequenced several times. Some of our 1000 Genomes Project read data, mainly Roche 454 read data obtained from the Baylor College of Medicine, were highly redundant (Figure 3.5). To prevent the duplicate fragments from skewing our SNP discovery, MosaikDupSnoop (Figure 2.15) was created to filter the duplicate paired-end reads. In an early batch of Roche 454 runs, an average of 36 % of the resolved paired-end reads were duplicates (Figure 3.6). In subsequent runs, modifications of sequencing library preparation reduced the average percentage of duplicates to 16 %. In the single-end runs, 5 % of the fragments were marked as duplicates (Figure 3.7). The difference between the improved paired-end and single-end duplicate fractions is most likely caused by inefficiencies during the mate-pair circularization and capture (discussed in Section 1.2.4). The single-end duplicate removal algorithm only removes reads that share the same endpoints, whereas a small tolerance of 2 bp in the endpoints is accepted when removing duplicates in paired-end reads. The tolerance allows for tiny differences in fragment length due to 454 reads starting or ending with homopolymers or due to local alignments skipping sequencing errors.

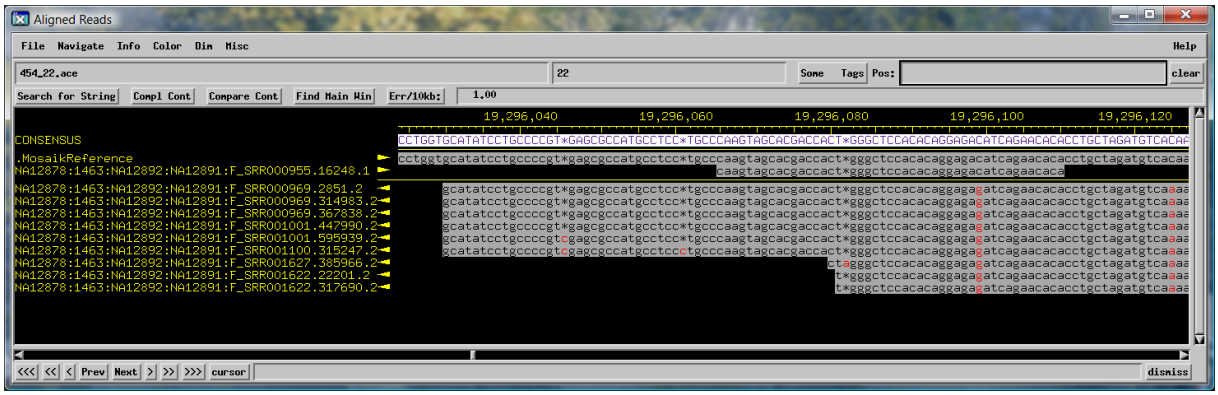

Figure 3.5. Duplicate Roche 454 reads. This figure shows one distinct fragment in the forward orientation and two (possibly three) fragments in the reverse orientation despite having ten reads.

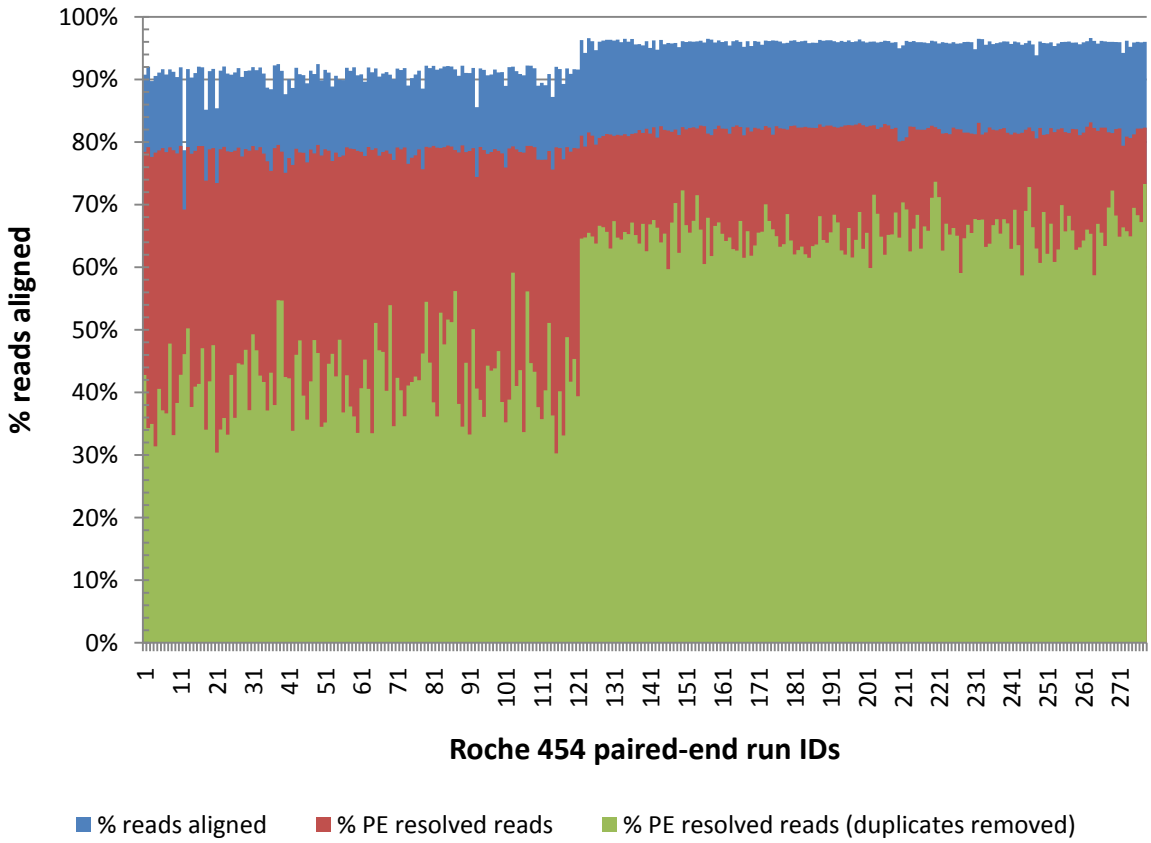

Figure 3.6. Duplicate removal in paired-end 454 runs. The percentage of concordant read-pairs that remain after duplicate removal (green).

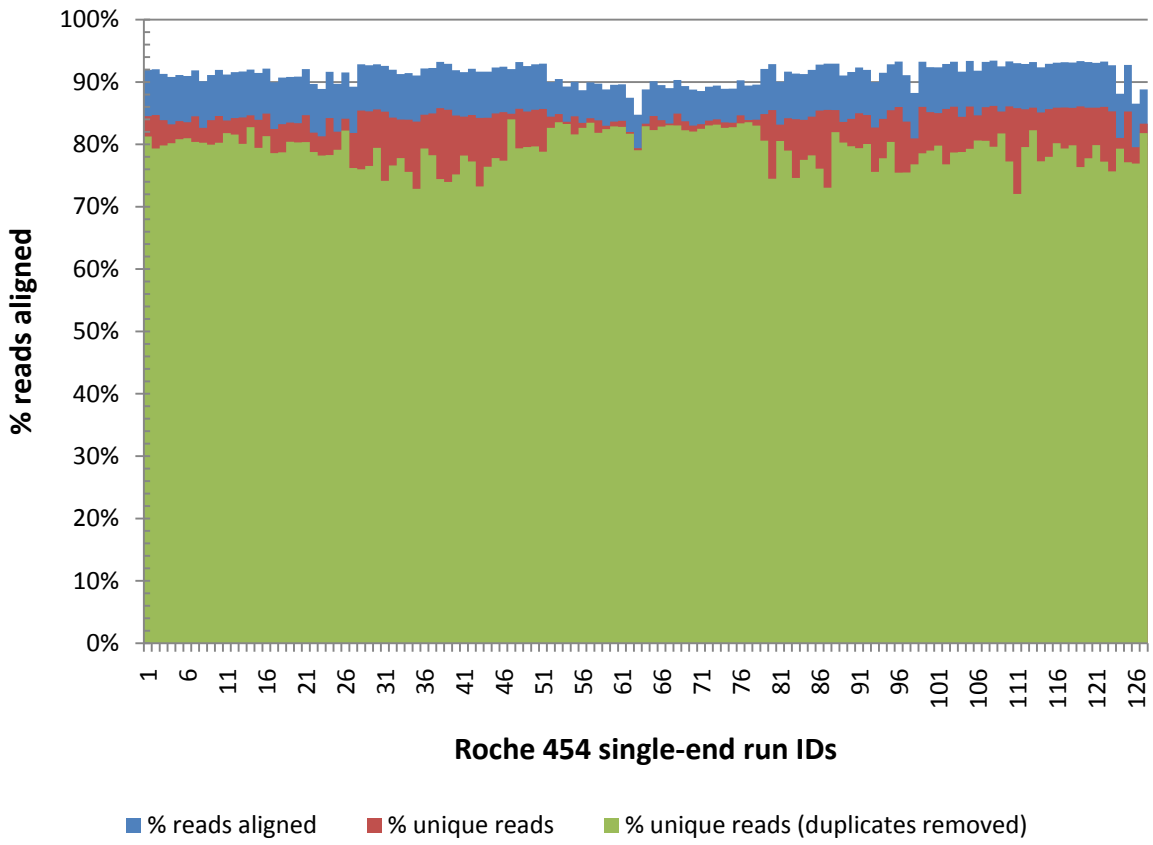

Figure 3.7. Duplicate removal in single-end 454 runs. The percentage of unique single-end reads after duplicate removal (green).

Read alignment

MOSAIK was used to align paired-end reads from both the Roche 454 and Illumina data sets. Up to four mismatches were allowed when aligning the Illumina reads and up to 5 % of the bases in a Roche 454 read were allowed to have a mismatch. The maximum number of reference sequence locations evaluated for each hash (mhp) was set to 100. This setting provides a performance boost at the cost of missing some alignments with highly repetitive hashes.

When resolving paired-end reads, we filtered out reads where both mates aligned non-uniquely. Reads where both mates are non-unique tend to occur in repetitive regions, where it becomes less likely that the proper pair of concordant

mates is represented in the discovered alignments. Including this class of reads tends to increase the overall alignment rate. However by excluding this class, the overall misalignment rate is reduced to 0.05 %.

After paired-end resolution, each run was screened for duplicates and merged into one alignment archive file. This file contained nearly 35x of overall genome coverage from the three trio family members. MosaikAssembler was used to create a multiple sequence alignment from the alignment archive and to export the results into the GigaBayes gig file format.

SNP discovery

GigaBayes was used to perform SNP and short-INDEL discovery using the gig files. This was the first time GigaBayes had been tested using a new trio-aware SNP discovery algorithm that takes advantage of the additional information about how samples are related to one another. The following filters were used with either GigaBayes or the SNP evaluation program:

1. Sites where P(SNP) (the posterior probability that a site is polymorphic) was less than 0.999 were discarded
2. Bases that had a quality score less than 10 were ignored
3. Sites where there were fewer than two reads for each allele were discarded
4. Sites where the minor allele was not observed on both strands were discarded
5. Sites that were within 12 bp of another SNP candidate were discarded.

SNP candidates that occurred within 12 bp of each other were less likely to overlap with SNPs in either dbSNP or WTSI SNP data sets (Figure 3.8). Visual inspection of the alignments revealed that clusters of nearby SNPs were often caused by systematic misalignments. Most of the misalignments had a minor allele that was present only on one of the strands. The mhp parameter was configured to use a maximum of 100 positions per hash. If the mhp parameter is set too low, the risk is that the positions needed to properly align a read will not be present. The alignments that had a minor allele only on one strand probably aligned better to other parts of the genome, but the aligner lacked the necessary information to find those locations.

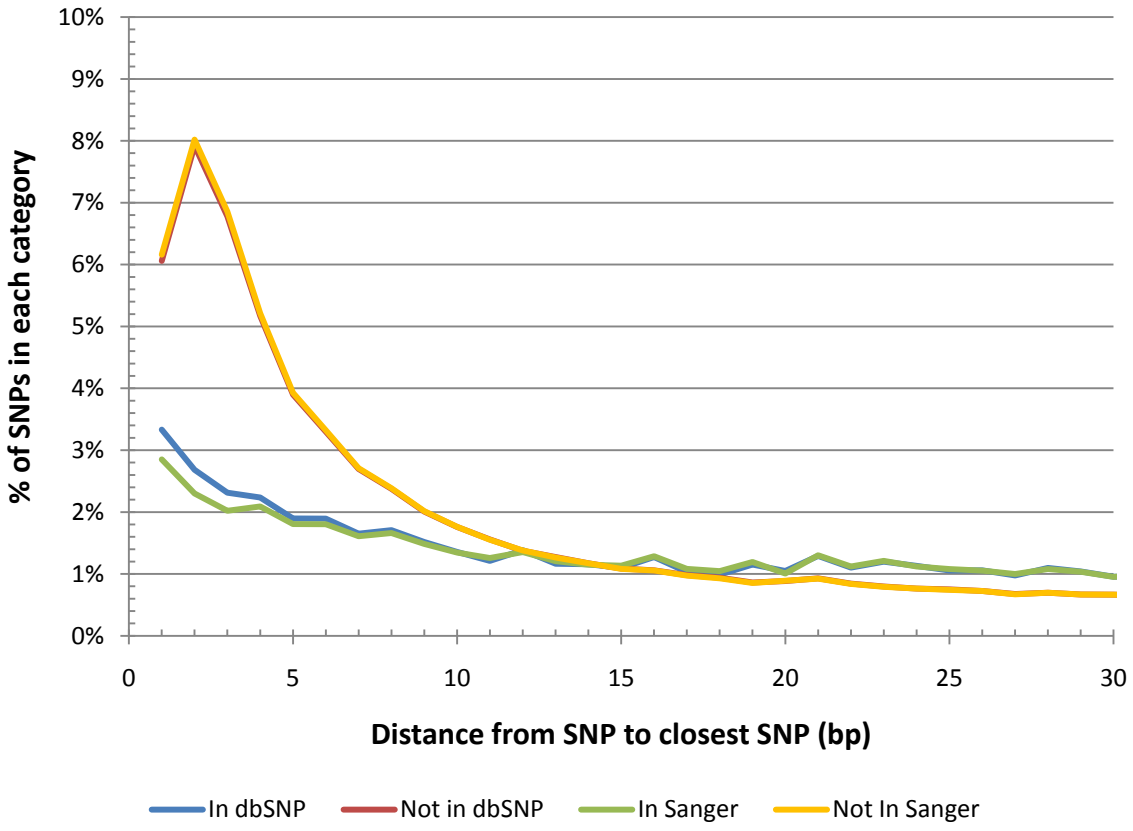

Figure 3.8. SNP candidates that occur close to each other tend to be associated with calls made only by GigaBayes.

A total of 3,396,969 SNP candidates were identified in our data set giving a mutation rate of one SNP per 907 bp between members of the trio. In comparison, the Wellcome Trust Sanger Institute (WTSI) data set features 4,496,207 SNP candidates which translates to a mutation rate of one SNP per 685 bp. Using only the released 454 and Illumina paired-end data, our aligned read coverage was roughly one half of the coverage featured in the WTSI data set. Besides differences in the alignment and SNP discovery algorithms used by each group, the difference in aligned read coverage seems to explain the observed decreased detection efficiency.

Two hundred SNPs were randomly chosen from the SNPs that were unique to our GigaBayes calls. These SNPs did not overlap with dbSNP (129), HapMap3, or the WTSI SNP calls. This strategy was chosen since the overlapping calls should have a high rate of validation, and we were curious how well the unique calls would validate. Correspondence with Richard Durbin revealed that his SNP caller considered 195 of those sites to be homozygous reference calls and the remaining five sites were filtered out for various reasons.

Our SNP calls were validated with a Sequenom SNP genotyping assay at the Broad Institute. The Sequenom assay is based on multiplex PCR followed by a single base primer extension reaction[116]. The extension products are then analyzed using MALDI TOF mass spectroscopy. Of the submitted 200 SNP candidates, 193 were successfully assayed. 33 SNP candidates validated giving us a 17.1 % validation rate in the SNP calls that were unique to GigaBayes. If all the overlapping SNP candidates

were to validate successfully, the upper bound of our validation rate would be at 87.5 %. Since we assume that a few of the SNPs that exclusively overlap with the WTSI data will fail to validate, the actual validation rate is probably slightly lower.

3.3.4. Re-alignment and INDEL calling

Bug fixes

Due to the high false positive rate observed in the SNP validation results, both the alignment and the SNP calling algorithms were scrutinized. During this evaluation, bugs were identified in both MOSAIK and GigaBayes. To make homopolymer INDELs line up properly in the multiple sequence alignment file, MOSAIK has a homopolymer correction algorithm that moves bases in a large insertion to the 5′ end and moves all the gaps to the 3′ end. Under some circumstances, this algorithm would fail and produce strange results that might induce false positive SNP candidates from GigaBayes. In addition, it was observed that false positives tend to occur in highly repetitive regions. To counteract this, the number of reference sequence locations evaluated per hash (mhp) was increased five-fold to 500. This improved MOSAIK's ability to determine if a read could be aligned uniquely or non-uniquely.

When visually analyzing the false positive SNP calls in the multiple sequence alignment, a pattern began to emerge. Usually, a SNP was observed on one strand but not the other strand. This behavior resulted from alignment artifacts. An option was added to GigaBayes in the previous SNP calling iteration to filter these events,

but it was implemented incorrectly. That version included an option where the read coverage for the minor allele on both strands was required to be greater than a user-specified threshold. After the bug fix, the new version checked that a minimum number of minor alleles were found on each strand. By forcing all SNP calls to have two minor alleles on each strand, 150 of the 160 false positive SNP calls that were assayed using Sequenom disappeared. The updated option effectively increased the minimum coverage from 4x to 8x, however the increase in required coverage was unlikely to be the mitigating factor since the median coverage for the CEU trio was 24x per individual.

Read alignment

MOSAIK was used to align all the reads in the pilot 2 Roche 454 and Illumina data sets. Using our nine node computational cluster, it took almost 9.4 days for the entire alignment and variant calling pipeline to process 6.1 billion reads (393 Gb or 128x total read coverage) (Figure 3.9). The overall alignment speed was a bit slower because of the large number of reference sequence locations evaluated per hash (mhp 500). No other modifications were made to the read alignment protocol.

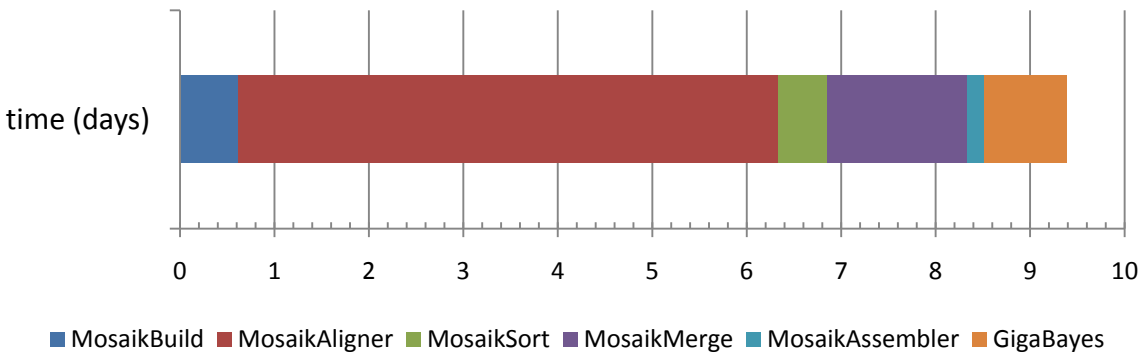

Figure 3.9. Read alignment and variant calling pipeline time. The figure shows the duration of time spent with each program in the pipeline. MosaikAligner, MosaikSort, and GigaBayes were executed in parallel on the cluster. The remaining programs were executed serially.

Short-INDEL discovery

Shortly after validating the previous iteration of SNP calls, the 1000 Genomes Project was interested in investigating how well the different pipelines perform when calling short-INDELs. To avoid the homopolymer artifacts that complicate Roche 454 alignments, a decision was made to only use the Illumina data set for short-INDEL discovery. The following filters were used with either GigaBayes or the SNP evaluation program:

1. Sites where P(SNP) (the posterior probability that a site is polymorphic) was less than 0.999 were discarded
2. Bases that had a quality score less than 10 were ignored
3. Sites where there were fewer than two minor alleles on each strand were discarded
4. Sites that were within 6 bp of another SNP candidate were discarded.

SNP candidates that occurred within 6 bp of each other were less likely to overlap with SNPs in either dbSNP or WTSI SNP data sets (Figure 3.10).

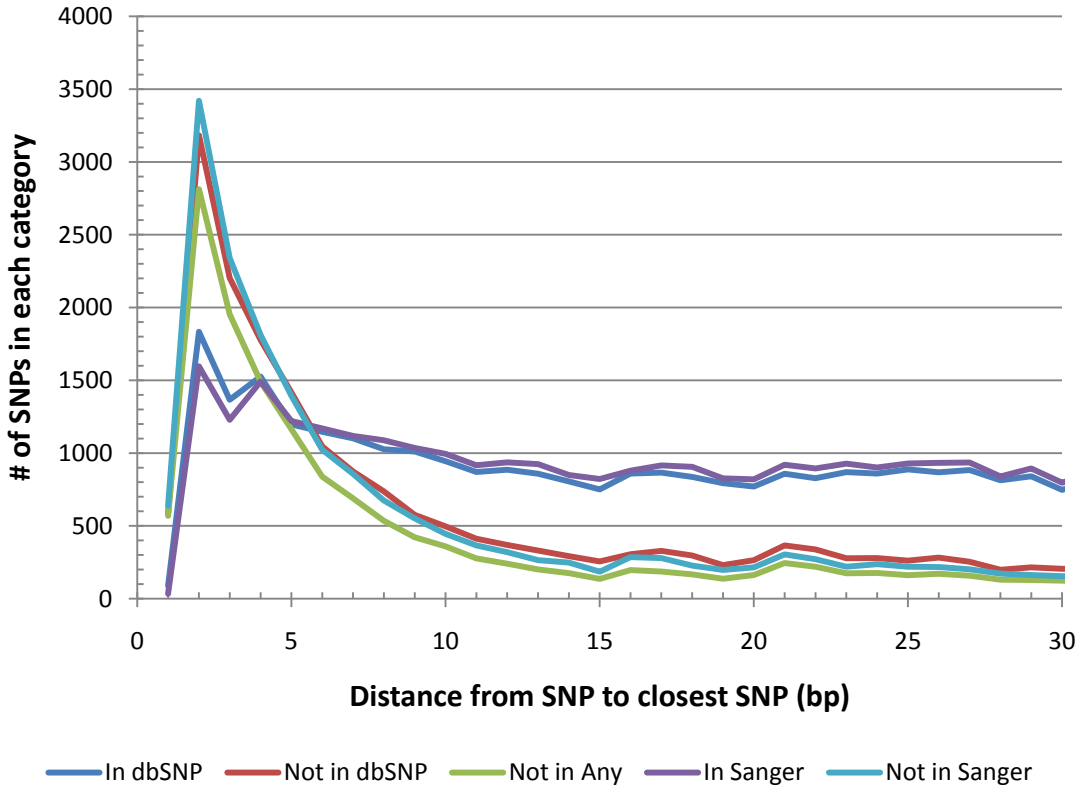

Figure 3.10. SNP candidates that occur close to each other tend to be associated with calls made only by GigaBayes.

A total of 3,427,445 SNP candidates and 441,003 short-INDELs were identified in our data set giving a mutation rate of one SNP per 899 bp and one INDEL per 7 kb. In comparison, the WTSI data set features 4,496,207 SNP candidates, which translates to a mutation rate of one SNP per 685 bp and one INDEL per 9.3 kb.

Gerton Lunter from the Department of Physiology, Anatomy and Genetics at the University of Oxford validated a random subset of 265 INDEL calls using PCR. A set of 14 of the 265 short-INDELs could not be confirmed using PCR, indicating a false discovery rate (FDR) of 5.3 %. The algorithms used by our pipeline and the

University of Oxford performed the best compared with the algorithms used at the National Human Genetics Research Institute (NHGRI), the Wellcome Trust Sanger Institute (WTSI), and Yale University. The University of Oxford performed slightly better when detecting 1 bp insertions and deletions, and we performed slightly better when detecting insertions and deletions in homopolymer regions (Figure 3.11).

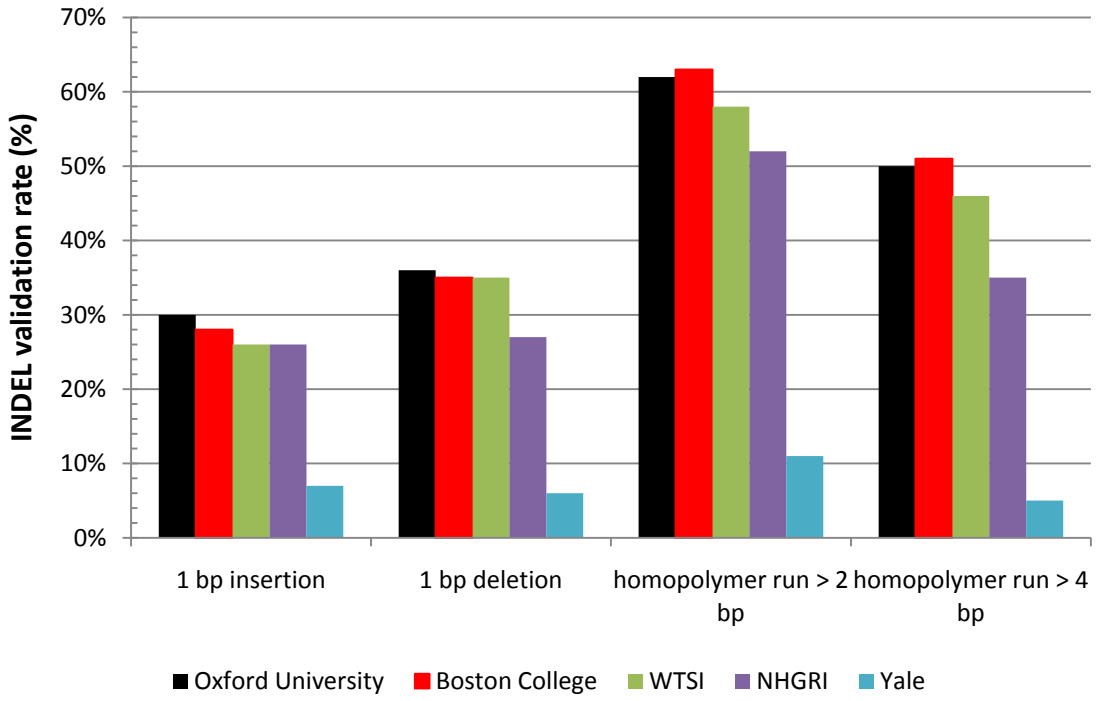

Figure 3.11. Short-INDEL validation results for 1 bp INDELs, as well as INDELs occurring in homopolymer runs.

3.3.5. Summary

Perhaps the greatest improvement in the alignment and variation calling pipeline occurred during the 1000 Genomes Project. MOSAIK was heavily modified to minimize unneeded steps in the alignment protocol. New tools were created to deal with duplicate reads in low complexity libraries, as well as in analyzing the output from GigaBayes. Crucial bugs in both MOSAIK and GigaBayes were identified and

remedied. These improvements have contributed to the successful results produced by the Marth Lab in subsequent 1000 Genomes Project analysis studies.

4. Mobile element insertion discovery

4.1. Introduction

Mobile elements (ME) are endogenous genomic sequences that transpose or retrotranspose into locations across the host genomes[117]. Most annotated MEs found in the human genome are no longer active; only a few types of mobile elements are responsible for most of the ME polymorphism among *Homo sapiens*. These include various subclasses of the 300 bp ALU elements, the 6 kb L1 elements, and the 1.5 kb SVA elements[118]. Mobile element insertions (MEI) are known to cause significant structural variation within *Homo sapiens*[119,120]. Furthermore, they are contributors to disease[121], are used in population studies, and have diverse functional impact[119,120,122].

MEI events are observed either as deletions or insertions when one genome sequence is compared with another. Mechanistically, both types of observations are due to insertions since precise excisions of MEs are rare[117] . The availability of two fully assembled human genomes[16,18] enabled the first genome-wide comparisons of mobile element polymorphisms to date[123,124], and the identification of novel loci containing mobile elements. Recently published genomes[90,125-128] based on reference-guided alignments on massively parallel sequenced short-read data have largely omitted the detection of mobile element insertions.

We introduce two methods that identify novel mobile element insertions. The first method uses a tool developed in our lab, Spanner[73], to perform MEI discovery in Illumina paired-end data sets (Figure 4.1A). The second method uses MOSAIK to

perform split-read alignments on Roche 454 data sets (Figure 4.1B). These methods were applied to the data sets from the low coverage and deeply sequenced trio studies in the 1000 Genomes Project. This chapter will focus on my work developing the split-read method. The Illumina read pair method will be described briefly, but will otherwise remain beyond the scope of this thesis. However, the results from both methods will be discussed with respect to validation rates and detection efficiency.

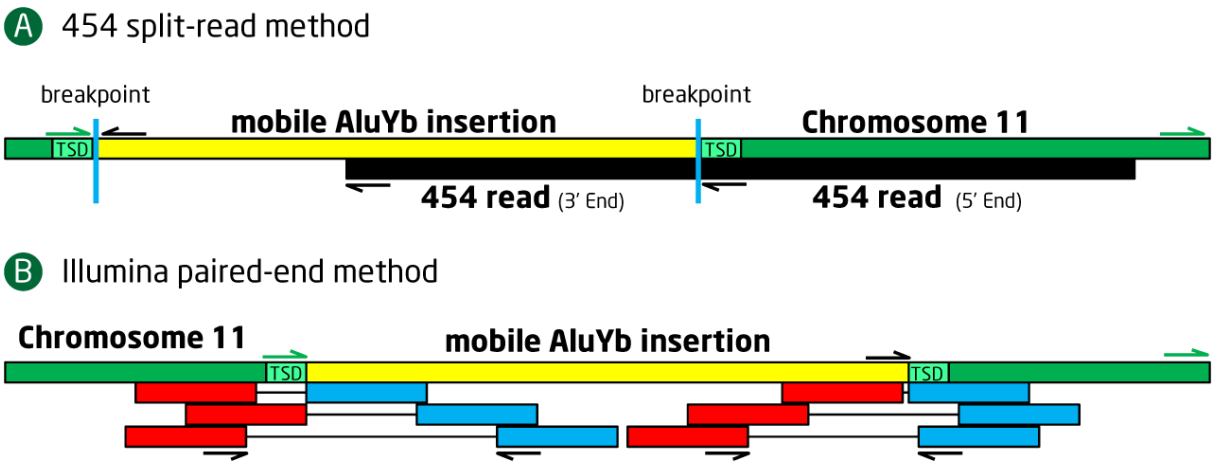

Figure 4.1. Mobile element insertion discovery methods with respect to the sample genome. (a) Roche 454 reads are aligned to a collection of transposable element consensus sequences and then the unaligned portion of the read is aligned back to the human genome. Each split-read alignment traverses at least one of the breakpoints in the mobile element insertion. (b) Illumina paired-end reads are aligned to both the human genome and the collection of transposable element consensus sequences. Our structural variation tool, SPANNER, uses these alignments to detect potential mobile element insertions. SPANNER detects discordant paired-end reads where one mate aligns uniquely to the genome and the other aligns to one of the transposable elements.

4.2. Data

4.2.1. 1000 Genomes Project data sets

Previously, SNP and INDEL calling was performed in the 1000 Genomes Project using the high coverage European trio dataset (Section 3.3). For this study, all the Roche 454 and Illumina reads from the low coverage (pilot 1) and high coverage

(pilot 2) studies were used. Pilots 1 and 2 represent different strategies for discovering genetic variants.

In Pilot 1, roughly 180 individuals have been sequenced at low coverage. It is problematic to detect a genetic variant in a single individual using a low coverage data set; however, the chance of detecting that genetic variant increases as more individuals are sampled. Twenty-two individuals were sequenced using the Roche 454 platform generating an average of 2.2x aligned read coverage for each individual. In contrast, 138 individuals were sequenced with the Illumina platform generating an average of 3.2x aligned read coverage for each individual.

In pilot 2, a European and a Yoruban family (trios) were sequenced to high coverage. Using deeply sequenced trios, the probability of finding most of the genetic variants in individuals is much higher than in pilot 1. On the other hand, it is less powerful than pilot 1 at elucidating genetic variants that persist in a given population. All six individuals for pilot 2 were sequenced on the Illumina platform generating an average of 24x aligned read coverage for each individual. In contrast, only the two children in each family were sequenced using the Roche 454 platform, generating an average of 8.7x aligned read coverage for each child.

4.2.2. James Watson data set

In addition to the 1000 Genomes Project data sets, the reads belonging to James Watson's genome[128] were downloaded from the Short Read Archive[129]. The main study originally had 7.4x read coverage, but only 19.2 Gb of reads (6.2x) were

available for download. Dr. Watson requested that all data surrounding or sharing haplotype structure with the ApoE gene (associated with late onset Alzheimer's disease) be redacted, but that would not account for nearly 1.2x coverage of missing sequence data.

4.2.3. Mobile element annotations

The consensus sequences for many subfamilies of Alu, SVA, and L1 mobile elements were extracted from RepBase[130] annotations. The locations where these mobile elements occur in the human genome (NCBI36 / hg18)[16] were extracted from the RepeatMasker[101] annotations. A mobile element reference sequence was created using the mobile element subfamilies found in RepBase. A total of 23 AluY, 6 AluS, 1 SVA, and 22 LINE1 mobile element consensus sequences were used in the mobile element reference sequence (Supplementary Figure 5).

4.3. Roche 454 split-read method

4.3.1. Aligning the data sets to the human genome

MOSAIK was used to align all the Roche 454 and Illumina reads from the 1000 Genomes Project pilots 1 and 2. The reads were aligned to a composite reference sequence consisting of both the human genome and the mobile elements consensus sequences. The jump database, used by MOSAIK to store hash locations in the reference sequence, was modified to prioritize alignments to the mobile element references. This guarantees that we will always know if a read aligns to one of the

mobile element subfamilies even if the reads were aligned with less sensitive alignment parameters. Similarly, the local alignment search option was used to rescue alignments in the Illumina paired-end data sets that might otherwise be missed by less sensitive alignment parameters. Also, MOSAIK was configured to store all reads that did not align to either the genome or the mobile element consensus sequences in a fastq file. These unaligned reads represent the subset of reads that are most likely to align to regions not represented in the reference genome.

4.3.2. Aligning the data set to the mobile elements

MOSAIK was used to align all the unaligned Roche 454 reads to the mobile element consensus sequences. We seeded the alignments with a 15 bp hash, allowed up to 5 % of the bases to have a mismatch with respect to the reference, and added the constraint that at least 40 bp of the read must align to the mobile element references. Approximately 0.5 % of the mate-pair reads and 0.9 % of the single-end reads aligned to the mobile element consensus sequences (Figure 4.2A).

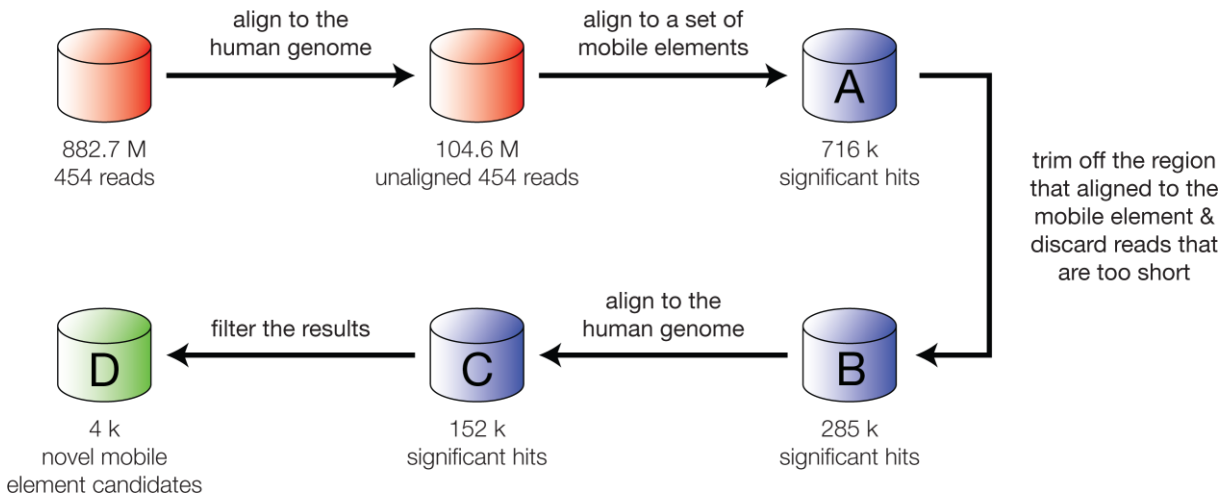

Figure 4.2. Overview of the Roche 454 split-read method. Starting with more than 100 million unaligned reads, roughly 4000 mobile element insertion candidates are found after split-read alignment and filtering.

4.3.3. Read trimming and aligning the data set to the genome

A program was created, MoblistTrimmer, which scans the alignments for each aligned read and records the target region where the read aligned to a mobile element. If multiple overlapping alignments are found, the target region is expanded to cover the overlapping alignments. More than 99.9 % of the aligned reads will have only one target region. In these cases, the unaligned sections before and after the target region are compared. The longer unaligned section is kept, while the remainder of the read is trimmed. If multiple non-overlapping target regions are found, the read is discarded. If less than 40 bp remained after trimming, the read was discarded. Twenty-five percent of the mate-pair reads and 44 % of the single-end reads from the previous step remained after trimming (Figure 4.2B).

The trimmed reads were aligned to the human genome using the sample alignment parameters that were used previously. Forty-six percent of the mate-pair

and 54 % of the trimmed singled-end reads aligned back to the human genome (Figure 4.2C).

4.3.4. Joining the split-read alignments

Another program was created, DetectNovelSplitReadEvents, which combines the results from both alignment runs and stores them in an MEI candidate file format. DetectNovelSplitReadEvents discards all reads that align to multiple locations in the human genome. The remaining reads are realigned to the human genome using sensitive alignment parameters: up to 9 % of the bases are allowed to have a mismatch with respect to the reference and the alignment must be at least 90 % of the original read length. The realignment helps identify reads that were initially unaligned because less-sensitive alignment parameters were used and therefore reduces the number of false positives. The aligned read names are then fed back into DetectNovelSplitReadEvents, which produces a filtered MEI candidate file.

4.3.5. Producing the MEI candidates for the Watson genome

The Roche 454 split-read method was also applied to the Watson genome. The protocol was modified to fit time constraints. Instead of first aligning the reads to the human genome and collecting the unaligned reads, the Watson reads were aligned directly to the mobile element consensus sequences. The realignment that takes place after the first pass of DetectNovelSplitReadEvents removes the split-reads that already occur in the human reference sequence (Figure 4.3A).

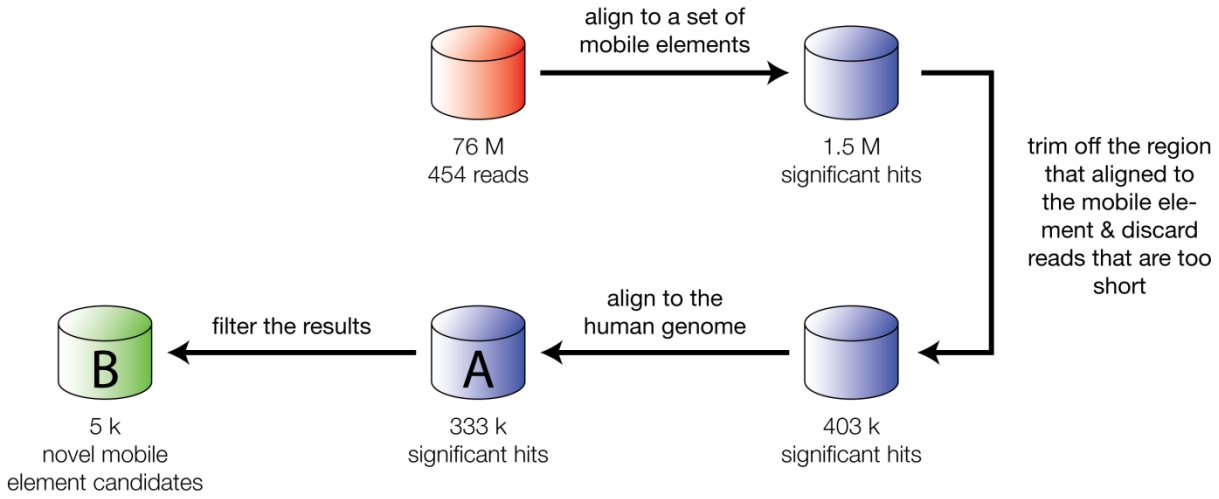

Figure 4.3. Applying the modified Roche 454 split-read method to the Watson data set. Starting with more than 76 million reads, roughly 5000 mobile element insertion candidates are found after split-read alignment and filtering.

4.3.6. Additional filtering and split-read clustering

In each aligned read, there can be up to three unaligned regions: a gap occurring before the genomic hit (genome gap), a gap occurring after the mobile element hit (mobile element gap), and a gap occurring between the genomic and mobile element hits (mid gap) (Figure 4.4).

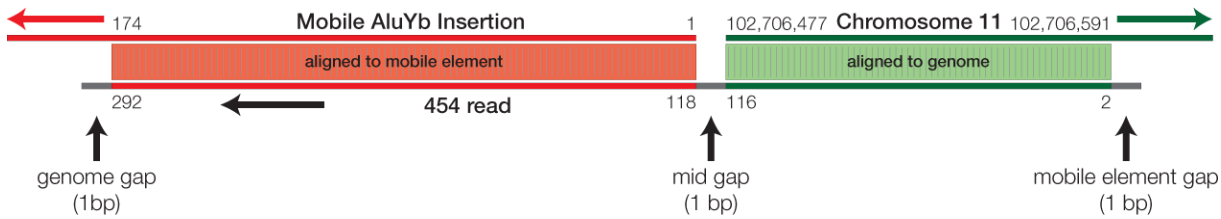

Figure 4.4. Split-read alignment. Additional filtering was performed on the MEI candidates based on irregularities in the alignment and proximity to annotated mobile elements.

Deniz Kural (Boston College) led the effort to reduce the number of false positives in the MEI candidates and to cluster them into MEI events. All split-reads where the mid gap was larger than 6 bp were filtered. Reads were also filtered when the genome or mobile element gaps were larger than 6 bp, except when the read was

long enough to contain the entire mobile element. These filters helped remove false positives at the risk of removing some split-reads that might have transduced DNA (DNA that has been moved from one region of the genome to another region of the genome) proximal to the MEI. Split-reads with an alignment quality less than 40 (greater than 0.01 % chance that the read was misaligned) and mobile element hits less than 60 bp were filtered. All split-reads that occurred within 100 bp of annotated Alus, L1s, and SVAs were also filtered. After filtering, only 2.8 % of the original MEI candidates remained from pilots 1 and 2 (Figure 4.2D), and 1.5 % of the original MEI candidates remained from the Watson data set (Figure 4.3B).

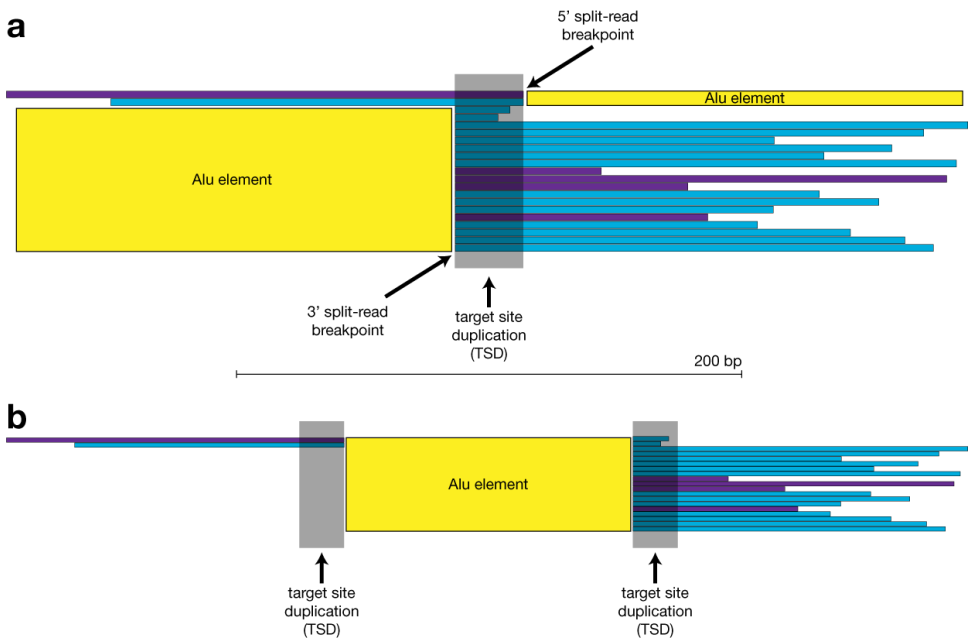

Figure 4.5. Split-read clusters. Gray boxes indicate the target site duplication (TSD) occurring at each breakpoint. Yellow boxes indicate on which side of the genomic alignment, the mobile element alignment can be found. Figure (a) shows the event in the reference genome and (b) shows the same event in the sample genome.

The split-reads were clustered together into MEI events. The minimum number of supporting reads was one, but most events had multiple supporting fragments with split-reads spanning both breakpoints (Figure 4.5).

4.4. Illumina paired-end method

Chip Stewart (Boston College) used his structural variation tool, Spanner[73], to identify MEIs in 1000 Genomes Project pilots 1 and 2 using Illumina paired-end reads. Spanner filtered out concordant read-pairs. MEI candidates were chosen from paired-end reads where one mate sequence aligned uniquely to the genome with an alignment quality greater than 40, and the other mate sequence aligned to the mobile element consensus sequences. After clustering the candidates, an MEI event was called if supported by two paired-end reads. MEI events occurring within 400 bp of annotated mobile elements were discarded.

4.5. Validation

4.5.1. Candidate events

A total of 9589 MEI events were detected individually from the Illumina paired-end and the Roche 454 split-read pipelines in the 1000 Genomes Project pilots 1 and 2. After consolidating the events from both methods, 5364 distinct MEI events were detected each being supported by candidates from an average of 16 individuals (Figure 4.6). Of the 5364 distinct MEI events, 633 overlap with loci found in previous studies[124,131,132], leaving 4731 novel loci (Figure 4.7). A total of 974 MEI events were detected in the Watson genome by Roche 454 split-read pipeline.

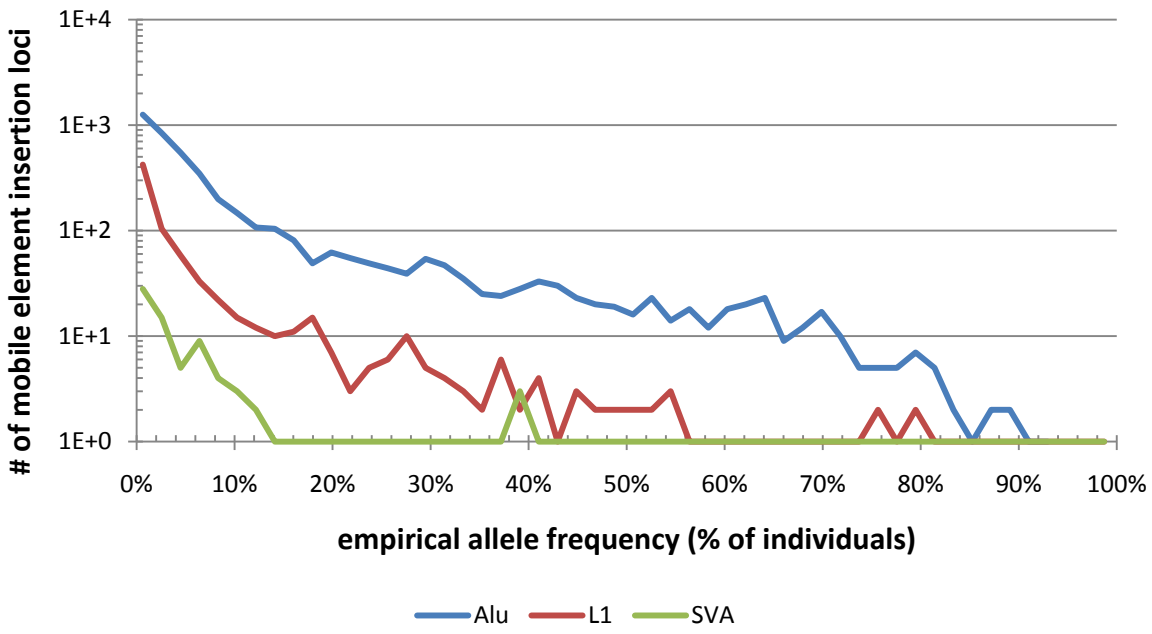

Figure 4.6. The empirical allele frequency spectrum for our MEIs in 156 individuals from both pilots 1 and 2. Since most of our data has low coverage (2 – 3x), genotyping heterozygous MEIs is near impossible. Rather than representing allele counts in a diploid organism, the empirical allele frequency spectrum shows the percentage of all individuals that have a specified number of MEIs.

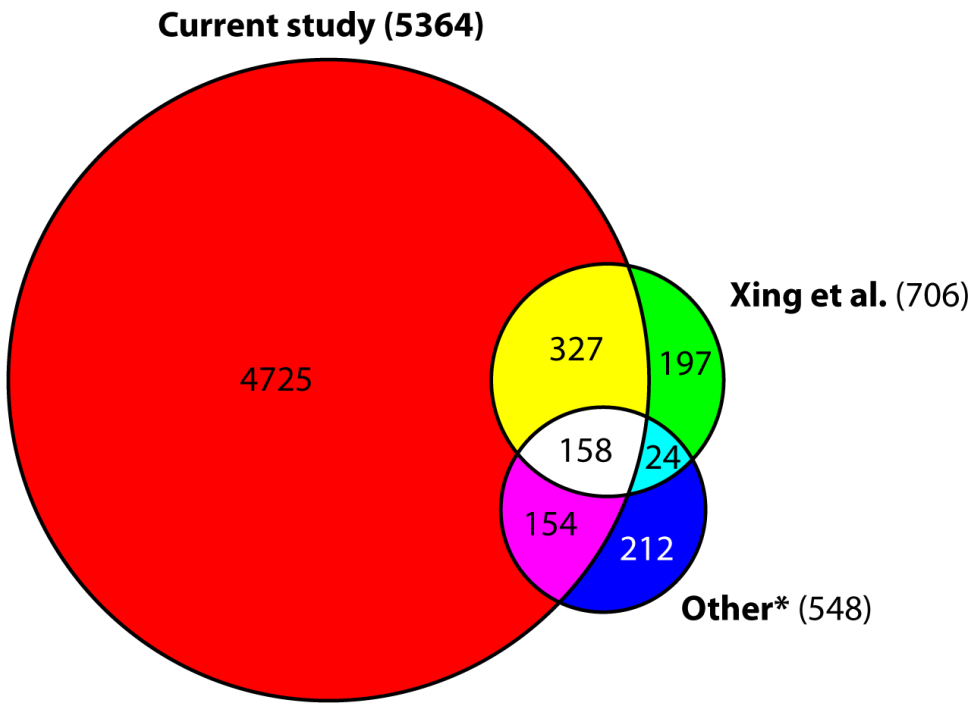

Figure 4.7. MEI event overlaps with the Xing *et al* study[124] and the MEIs contained in the online dbRIP* database[131].

4.5.2. Validation results

A random subset of 746 MEI events was chosen from the 5364 distinct MEI events for PCR validation to estimate false discovery rates (FDR) for our detection algorithms. The FDR for the Roche 454 split-read method was 3.3 % in pilot 1 and 5.4 % in pilot 2 with an average FDR of 4.4 %. Similarly, the FDR for the Illumina paired-end method was 4.5 % in pilot 1 and 2.2 % in pilot 2 with an average FDR of 3.7 %. The FDR of both methods were within the Poisson error fluctuations of each other (Figure 4.8 and Figure 4.9). In pilot 2, the paired-end method had a lower false discovery rate, which is probably due to the high coverage (~24x per individual) found in that data set.

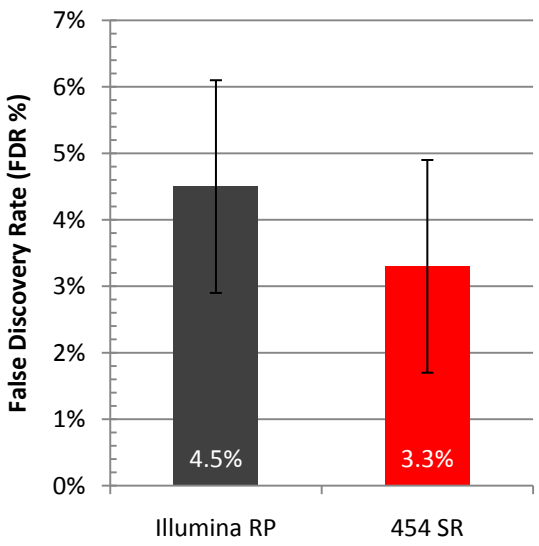

Figure 4.8. The false discovery rates of both methods in the pilot 1 study (low coverage with many individuals).

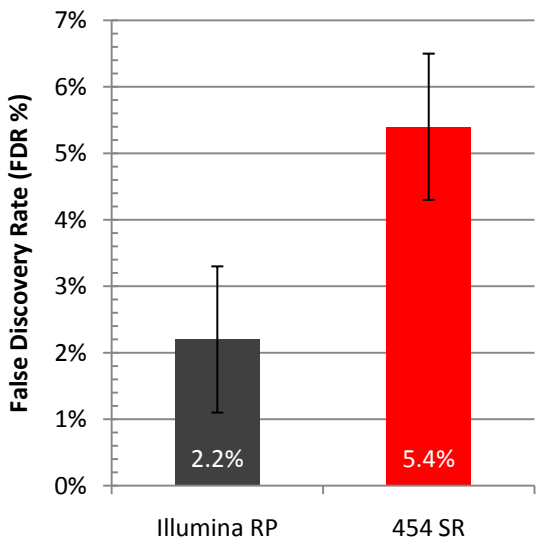

Figure 4.9. The false discovery rates of both methods in the pilot 2 study (high coverage in two trios).

When combining the validation results between the two methods, Alus have an FDR of 2.2 %, L1s have an FDR of 18.1 %, and SVAs have an FDR of 27.0 %. The overall combined false positive rate was 4.5 %.

In addition to the random subset of MEI events, other events were selected for validation experiments to confirm SVA insertions, unique MEI events in the CEU trio, and exon-interrupting events. A total of 850 MEI events were validated with PCR by our colleagues at Louisiana State University, Yale University, and the European Molecular Biology Laboratory (EMBL) in Heidelberg. The MEI events that were unique to the Watson genome were not validated since a public Watson DNA sample or cell line is not available.

4.6. Analysis

4.6.1. Detection efficiency in the trio children

Chip Stewart (Boston College) analyzed the detection efficiency of each detection method, as well as the combined detection efficiency. The number of events detected (N_{det}) are a composite of the true positives ($N_{det|t}$) and the false positives (N_{det}*FDR) (1). The detection efficiency (ε) is therefore the ratio of true positives to true events (2). The number of MEI events called in an individual can be expressed in terms of the FDR, detection efficiency, and number of true events for each method (3). N_{RP} is the number of MEI events called using the Illumina paired-end method and N_{SR} is the number of MEI events called using the Roche 454 split-read method.

$$N_{\det} = N_{\det|t} + N_{\det} \cdot FDR \quad (1)$$

$$\varepsilon = \frac{N_{\det|t}}{N_t} \quad (2)$$

$$N_{RP} = \varepsilon_{RP} \cdot \frac{N_t}{1 - FDR_{RP}}, \quad N_{SR} = \varepsilon_{SR} \cdot \frac{N_t}{1 - FDR_{SR}} \quad (3)$$

Solving for the read-pair and split-read detection efficiencies would not help at this point since we do not know how many true events exist. Using equation (3) as a template, the number of MEI events called by the combined methods can be calculated (4). Using equation (4), we can solve for the read-pair and split-read detection efficiencies (5). The false discovery rate events that are called by both methods is essentially 0, therefore we can safely neglect the FDR_{SR*RP} term.

$$N_{SRRP} = \varepsilon_{SR} \cdot \varepsilon_{RP} \cdot \frac{N_t}{1 - FDR_{RPSR}} \quad (4)$$

$$\varepsilon_{RP} = \frac{N_{SR \cdot RP}}{N_{SR}} \cdot \frac{1 - FDR_{SR \cdot RP}}{1 - FDR_{SR}}, \quad \varepsilon_{SR} = \frac{N_{SR \cdot RP}}{N_{RP}} \cdot \frac{1 - FDR_{SR \cdot RP}}{1 - FDR_{RP}} \quad (5)$$

Using these equations, the detection efficiencies for each method can be calculated with respect to the mobile element class (Table 4.1). The detection efficiency for the combined methods (ε_{RP*SR}) in the trio children is 90 % for Alus, 60 % for L1s, and 50 % for SVAs.

Table 4.1. The detection efficiencies of the Illumina paired-end method and the Roche 454 split read method for each mobile element class. NA12878 is the child in the European trio and NA19240 is the child in the Yoruban trio.

	Individual	N_{RP}	N_{SR}	N_{RP+SR}	ε_{RP}	ε_{SR}
Alu	NA12878	733	817	570	71% ± 4	79% ± 5
	NA19240	932	675	473	71% ± 4	51% ± 3
LINE1	NA12878	80	109	31	29% ± 13	40% ± 9
	NA19240	92	80	30	40% ± 20	33% ± 7
SVA	NA12878	11	6	2	70% ± 40	40% ± 30
	NA19240	27	1	0	-	-

4.6.2. Classifying MEI events

Of the 5364 MEI events, 5362 unambiguously belong to one of the three major mobile element classes (Alu, L1, SVA) based on the supporting MEI candidates. Of the 4496 Alu MEI events, more than 95 % belong to the AluY subclass. Of the 789 L1 MEI events, more than 90 % belong to the L1HS subclass.

4.6.3. Quantifying the number of ME events between two individuals

Based on a previous study, it has been estimated that 11k mobile element (ME) events occur between the human reference genome and the chimpanzee reference genome[133]. Carlos Bustamante (Department of Genetics at Stanford University School of Medicine) recently presented results indicating that Yoruban and European populations diverged more than 700 Kya. Based on molecular data, the divergence between humans and chimpanzees has been estimated to occur roughly 6 - 7 Mya[134]. Using a constant molecular clock, we would expect roughly 1150 ME events between a Yoruban and European child.

Using the detected ME event counts, the false discovery rates, and the detection efficiencies, Chip Stewart calculated the number ME events ($N_{A\Delta B}$) that occur between two individuals (A and B) (6). We have determined that 3255 ± 311 ME events occur between the Yoruban and European child (Table 4.2). Our results are therefore within a factor of two of the constant molecular clock estimation above.

$$N_{A\Delta B} = N_A \cdot \frac{1-FDR_A}{\varepsilon_A} \cdot \frac{1-FDR_B}{\varepsilon_B} - \frac{N_{A \cdot B}}{\varepsilon_A \cdot \varepsilon_B} \quad (6)$$

Table 4.2. The number of mobile elements estimated between two individuals with respect to mobile element class. * deletions were estimated using Spanner in the Illumina paired-end method.

	ME insertions*			ME deletions*			Total
	NA12878	NA19240	Relative	NA12878	NA19240	Relative	
Alu	980	1134	1840±210	801	942	1010±200	**2850±300**
LINE1	158	143	270±70	78	92	100±30	**370±80**
SVA	15	28	30±25	19	24	25±10	**55±30**

4.6.4. Investigating the overlap of MEI events in the European trio

Using the MEI events detected with the Illumina paired-end method, the Alu and L1 overlaps among the European trio family members were investigated. Most of the MEI events identified in the child (NA12878) were also identified in the parents (NA12891 and NA12892) (Figure 4.10 and Figure 4.11)

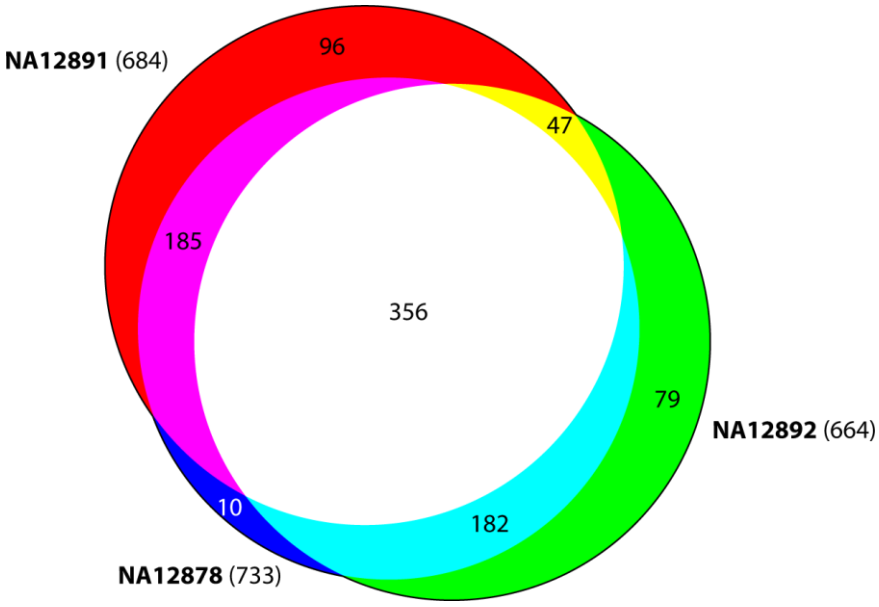

Figure 4.10. Alu MEI overlaps between the European trio family members.

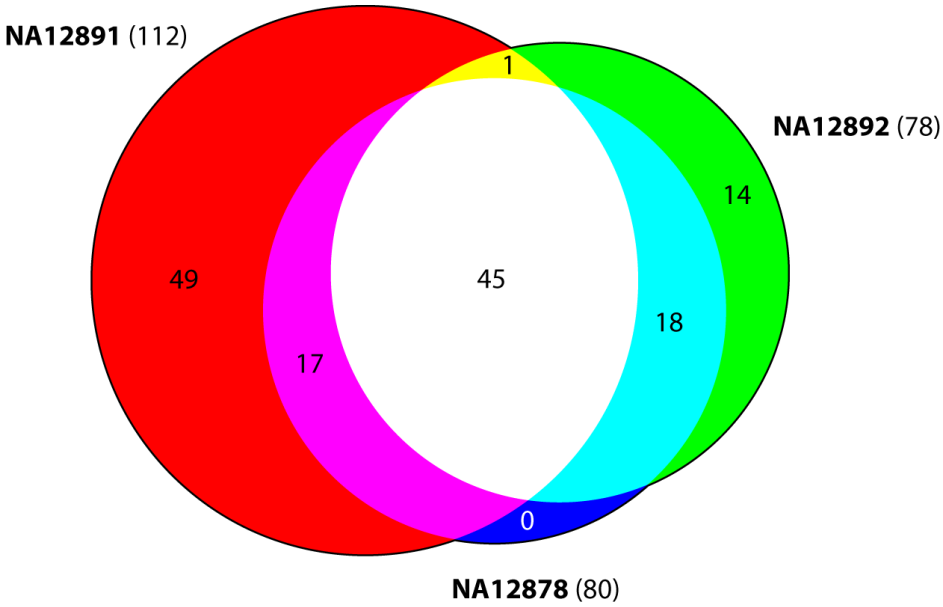

Figure 4.11. L1 MEI overlaps between the European trio family members.

The 10 Alu MEI events that were private to the child were submitted for validation using PCR. The validation determined that all 10 events existed in the child and that at least one of the parents also shared those events. Those events were a result of a failure to detect the event in the parents rather than evidence of a *de novo* MEI event.

4.6.5. MEI event overlaps with gene annotations

Many studies have shown that MEIs that occur in or proximal to genes have been associated with human disease[135-137] and some disease-causing MEIs have been known to inactivate genes[138,139]. We used GENCODE annotations[140] (v3b) to determine how many of our MEI events overlap with gene annotations (Table 4.3).

Table 4.3. MEI event overlap with GENCODE annotations.

ME class	Gene	Exon
Alu	1950	66
L1	356	19
SVA	41	1
Total	2347	86

The large number of MEI events that overlap the gene annotations, but not the exon annotations, suggests that many MEI events may be intronic. This is especially interesting considering that intron size has been associated with alternatively spliced genes and constitutively active genes[141]. MEI events that do not overlap with gene annotations suggest that many also occur in intergenic regions.

4.6.6. Investigating the MEI population clusters

To investigate how the MEI events cluster with respect to individuals, we created feature vectors out of the 375 PCR validated MEI events that occurred in the 25 assayed individuals from pilot 1. Principle component analysis was then performed on these feature vectors and the first two principle components were used as XY-coordinates (Figure 4.12).

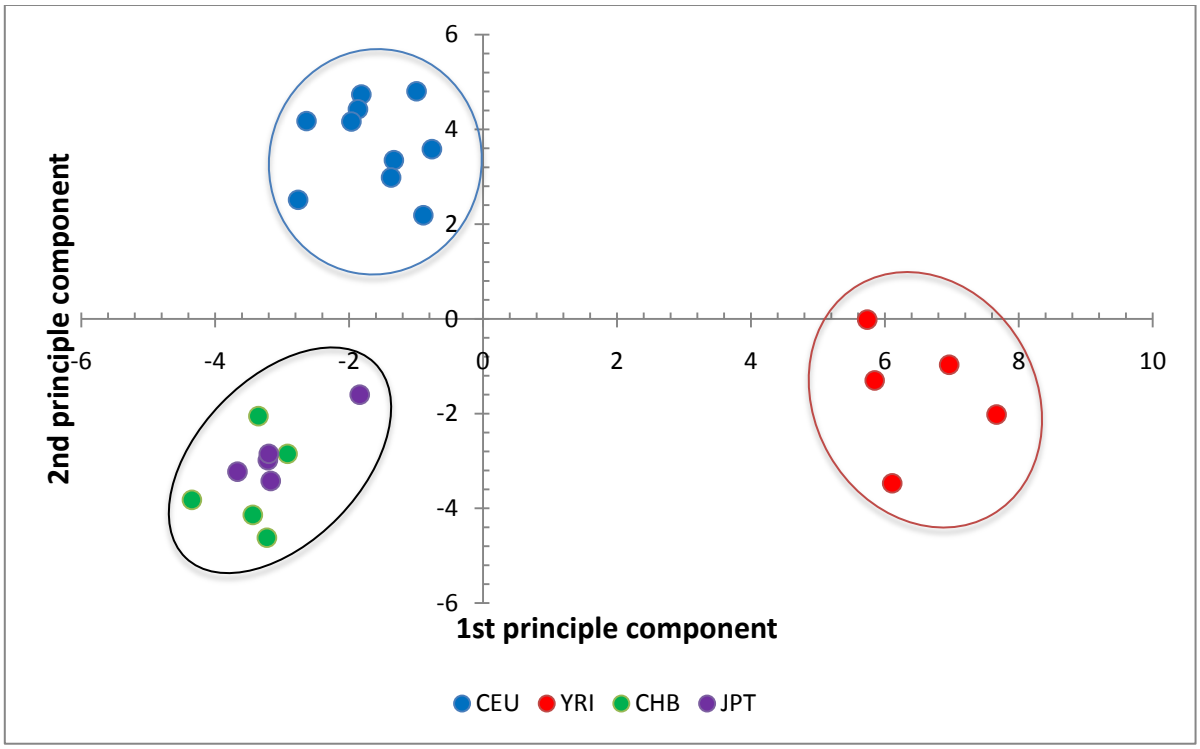

Figure 4.12. Principle component analysis was used to elucidate clustering of 375 MEI events for 25 individuals (circles). After clustering, circles were color coded according to population: Europeans (blue), Yorubans (red), Chinese (green), and Japanese (purple).

The results from the principle component analysis feature three well-segregated clusters that correspond to the three major HapMap[66] populations (European (CEU), Yoruban (YRI), and Chinese/Japanese (CHB/JPT)). Genetic markers such as SNPs have been shown to segregate similarly in the HapMap populations[66]. This suggests that MEI events behave similarly to other genetic markers when analyzing human populations.

4.7. Summary

Perhaps the most significant outcome of this study is that our group was able to infer that two individuals of distant ancestry differ by approximately 3255 ME events. This represents the first time a group has been able to make such a qualified estimate and

provide a breakdown of how many Alus, L1s, and SVAs are expected between two individuals. With 5364 detected MEI events, our study represents the largest catalog of such events to date[124,131]. The low false discovery rates measured in the Roche 454 split-read and Illumina paired-end methods are promising and indicate that most detected MEI events are probably true.

We are investigating ways of improving the sensitivity of our methods. The detection efficiency of our methods on Alu MEI events was high, yet the detection efficiencies on L1 and SVA MEI events were a bit lower. Alus almost never have transduced DNA associated with their insertions, but L1s and SVAs often have transduced DNA that follows them upon insertion. As a result, improved methods are being evaluated that improve the detection efficiency for MEIs that have transduced DNA. For example, the poly-A sequence on the 3′ ends of novel insertions may be different from the consensus sequences found in RepBase. The difficulty of base calling long homopolymers in 454 sequencing might also affect the length of the 3′ poly-A sequence. Both of these situations indicate that the strategy of filtering according to genome gap, mid gap, and mobile element gap should be updated to improve detection of MEIs with transduced DNA.

Like many scientific endeavors, this study raises more questions than answers. Our group is looking into the possibility of running RNA-Seq experiments to find mobile elements actively undergoing transcription. We are also experimenting with different techniques of dating the various MEI events using data sets with interesting

divergence times compared with modern human populations[142,143]. Finally, we are interested in using transduced DNA sequences to look for patterns where MEIs originate and where MEIs accumulate in the human genome.

5. Concluding Remarks

The sequencing technologies that have emerged on the market are already changing the landscape of computational biology. In the past several years, next generation sequencing platforms have enabled methods such as ChIP-Seq, RNA-Seq, target-capture sequencing, whole genome methylation studies, and structural variation discovery[23]. These methods have in turn enabled labs to perform experiments that were either too costly or impractical just a few years ago. The challenge is to identify key trends in sequencing technology and predict how these trends will enable or affect the research conducted in the Marth Lab.

5.1. Upcoming sequencing technologies

At the Advances in Genome Biology and Technology (AGBT) meeting this year, attendees were excited about all the new sequencing technologies that were being announced. Applied Biosystems released their SOLiD 4 platform and Illumina released their HiSeq 2000 platform. Both were essentially iterative updates that announced improvements in sequencing throughput and slightly lower error profiles. Most of the excitement, however, revolved around sequencing technologies offering longer reads and sequencing technologies offering low cost runs.

5.1.1. Ion Torrent

The original founder of 454 Life Sciences, Jonathan Rothberg presented the Ion Torrent sequencer. The technology builds on the observation that whenever a

nucleotide is incorporated by a DNA polymerase, a H^+ ion is released. Similar to the Roche 454 platform, nucleotides are introduced one at a time to an ION semiconductor chip that has been prepared much like a picotiter plate. Every time one or more nucleotides are incorporated, the pH level varies accordingly. Early results show the system has no problem distinguishing between homopolymers up to 6 bp long. The price point is what made the system appealing. The machine will cost less than $50k and each run will cost less than $500.

5.1.2. Life Technologies single-molecule sequencing platform

Life Technologies acquired VisiGen Biotechnologies back in 2008. Since then they have been working on a single-molecule sequencing platform that tethers 10 nm quantum dots to a DNA polymerase. The quantum dots are comprised of a CdSe core and a ZnS outer shell and when illuminated by a TIRF laser, provide the DNA polymerase with ATP for the next incorporation. They refer to the quantum dot-DNA polymerase constructs as sequencers. It was mentioned that after a while, the laser contributes to the degradation of the DNA polymerase which occurs after a fundamental read length of 1.0 – 1.5 kb has been reached. The interesting bit is that sequencers can be washed away and replaced with a new batch that will continue where the previous sequencers left off. It was unclear how much sequence data can be produced per run or how many times the sequencers can be exchanged. The sequencing technology is expected to be evaluated by early access customers at the end of the 2010.

5.1.3. Pacific Biosciences SMRT

The Pacific Biosciences platform uses a plate filled with zero-mode waveguides containing a DNA polymerase at the bottom of each waveguide. Fluorescently labeled nucleotides are detected during incorporation by the DNA polymerase. Approximately 80k reads are produced per run and they conform to an exponential read length distribution. It was mentioned that the tail end of the distribution sometimes show 10 - 20 kb reads. Similar to Helicos, dark bases result when the duration of the fluorophore emission is less than the detection interval. The SMRT machines are being delivered to early access customers during the summer of 2010.

5.2. High performance computing

As sequencing technologies generate more data, the analysis pipeline will become a larger bottleneck. Many labs are investigating alternatives to purchasing large computational clusters. In this regard, general purpose computation on graphics processing units (GPGPU)[144] and cloud computing[145] have recently been used in a few computational biology labs.

Back in 2007, NVIDIA created a parallel computing architecture called CUDA (compute unified device architecture). CUDA allows programmers to use the processing power available from modern high-performance graphics cards to speed up computationally expensive tasks. When a well-known reference-guided aligner, MUMmer, was modified to use a $130 graphics card, it achieved a 10x speedup when compared with the serial CPU version[146]. NVIDIA has also created the Tesla desktop

supercomputer which offers the 250 times the performance of standard PC for a hefty $10k. It would be an interesting experiment to move the pairwise alignment code in MOSAIK to the graphics processing units and see what sorts of gains can be made.

The Amazon Elastic Compute Cloud (EC2)[147] is currently the most popular commercial cloud computing service. Amazon charges an hourly fee based on what sorts of nodes are needed. For high-memory instances equipped with 34.2 GB of RAM and four processor cores, the hourly fee is currently $1.20 per hour. If MOSAIK's memory footprint can be reduced to under 7.5 GB RAM, an instance with two processor cores will cost $0.34 per hour. If additional persistent storage is needed, the Amazon Simple Storage Service (S3)[148] offers disk storage for $153 per TB per month. Finally the network bandwidth between the cloud and the outside world costs $0.15 per GB transferred. While this price structure might not be ideal for huge genome centers, it allows smaller labs to significantly increase their computational power at a moment's notice.

Besides public cloud facilities such as Amazon EC2, cloud computing can be implemented in a department using the open source Hadoop[149] software. Hadoop offers the MapReduce[150] software framework which allows applications to effectively apply the divide and conquer computational technique. Some bioinformatics tools and languages, such as CloudBurst[151], the Genome Analysis Toolkit[55], and BioPython[152], already incorporate the MapReduce algorithm.

With respect to MOSAIK, MapReduce could be implemented by dividing the genome equally between computational nodes, performing pairwise alignments on those nodes, and using the reduce operation to collect the results from each node. GigaBayes/BamBayes could map a set of reference locations to a set of computational nodes and use the reduce operation to collect the set of SNP and short-INDEL calls.

The Hadoop file system[153] (HDFS) is a distributed file system that can make these sorts of analyses more practical. HDFS is a fault-tolerant file system that ensures that data exists on at least three nodes. By using parity blocks, HDFS reduces the overhead of this replication from 3x to about 2x. The advantage of HDFS is that it scales better than the Network File System (NFS) protocol used in most UNIX environments. When 30 computational nodes need to access the same data, it may be more efficient to use a distributed file system.

5.3. Challenges to MOSAIK development

The newer sequencing technologies are offering in some cases much higher throughput than before and in other cases offering very long reads. The challenge is to modify MOSAIK to handle both extremes. MOSAIK currently pairwise aligns reads using the Smith-Waterman algorithm. The computational complexity of the Smith-Waterman algorithm makes long read alignment prohibitively slow. Alternatives such as implementing a BLAT- or BLAST-like algorithm should be considered. The current version of the Pacific Biosciences read aligner uses a suffix-tree approach that is similar to the MUMMER algorithm to quickly align long reads.

One of the caveats with single-molecule sequencing technologies such as the Pacific Biosciences SMRT or Helicos Heliscope platform is that bases are sometimes not registered and therefore show up as missing or dark bases. The current generation of gapped aligners uses gap penalties that are based on biological approximations on how often an insertion or deletion is expected. To be effective, two parallel gap scoring mechanisms will be needed. One that handles dark and missing bases and another that handles biologically relevant insertions and deletions.

Further emphasis on making MOSAIK a user-friendly platform will also be required. One idea is to use an automatic data set analysis tool to explore a data set of reads to determine the optimal alignment parameters. By examining the base quality distribution, the read length, and the known error profile for that sequencing technology, the analysis tool would be able to suggest a certain number of mismatches be used when aligning the data. This functionality can be extended to automatically detect the presence of sequencing adaptors that should be trimmed away from the reads.

Cloud computing and GPGPU programming was discussed in Section 5.2. However, pairwise alignments can be made much faster using normal processors by techniques such as single instruction, multiple data (SIMD)[154,155] processing. With some extra effort, programs can exploit the parallel nature of the processor instruction pipelines using SIMD instructions (called intrinsics) in C++ code. Some Smith-Waterman implementations have already seen up to 18x improvements in

alignment speed by using these SIMD intrinsics[156,157]. Even a 10-fold improvement in alignment speed would be remarkable in MOSAIK. The caveat is that these implementations only provide the forward Smith-Waterman algorithm (enough to produce a score), but do not feature the backtrace functionality required to reproduce the pairwise alignment.

The current alignment quality models were based on Roche 454 and Illumina reads. Roche 454 reads have more insertion and deletion errors than substitution errors. Illumina reads have more substitution errors than insertion and deletion errors. These models are then applied to other sequencing technologies that exhibit similar error profiles. Ideally, customized models should be generated for each sequencing technology.

5.4. Conclusion

MOSAIK has been used extensively in our lab to provide alignments for our genetic variant discovery studies. In our work with the 1000 Genomes Project Consortium, we have produced some of the best SNP, short-INDEL, and structural variation results, despite competing with groups with more resources. Many of the improvements made in MOSAIK have been the result of working in demanding analysis projects such as the *C. elegans* study, the *P. stipitis* study, and the 1000 Genomes Project pilot 2 study. With each development cycle, MOSAIK becomes more robust, which leads to more research labs adopting it for their experiments. I want to stress that leading analysis sessions during sequencing workshops has been

just as important for the development of MOSAIK as our own internal analysis studies. At these workshops I learn how other groups are trying to analyze their data. Often some minor changes in my code can make a huge difference for an entire research group.

Supplementary Figures

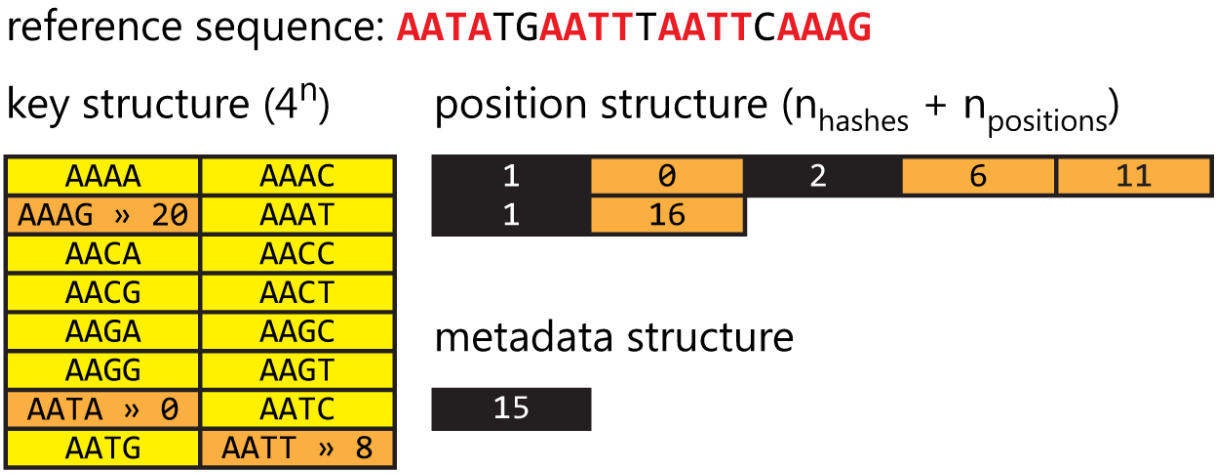

Figure S1. Jump database data structure. In this example, the reference sequence is divided into overlapping 4 bp hashes. The key structure is a sparse matrix where each slot in the key structure contains a 5 byte file offset for the position structure. The key database uses $5 * 4^{hash\_size}$ bytes of memory. To find the correct offset in the key structure we use the following equation:

$$key_{\text{offset}} = 5 \cdot hash_{\text{2-bit}}$$

For example, the hash AAAG in two-bit notation equals the number 3 when converted to a number. Jumping to byte 15 (5 * 3) in the key structure reveals that we can find the appropriate reference sequence locations at byte 20 in the position structure.

The hash AATT in two-bit notation equals the number 16 when converted to a number. Jumping to byte 80 (5 * 16) in the key structure reveals that we can find the appropriate reference sequence locations at byte 8 in the position structure.

The position structure is tightly packed and uses $4 * (n_{hashes} + n_{positions})$ bytes of memory. The first unsigned integer reveals the number of positions that are associated with that hash. The positions (unsigned integers) are given in the aggregate coordinate system where positions on chromosome 2 will have coordinates larger than the positions on chromosome 1.

The metadata structure contains an unsigned byte indicating the hash size used when creating the jump database. The aligner checks that the hash size specified by the user is the same as the hash size that was used when creating the jump database.

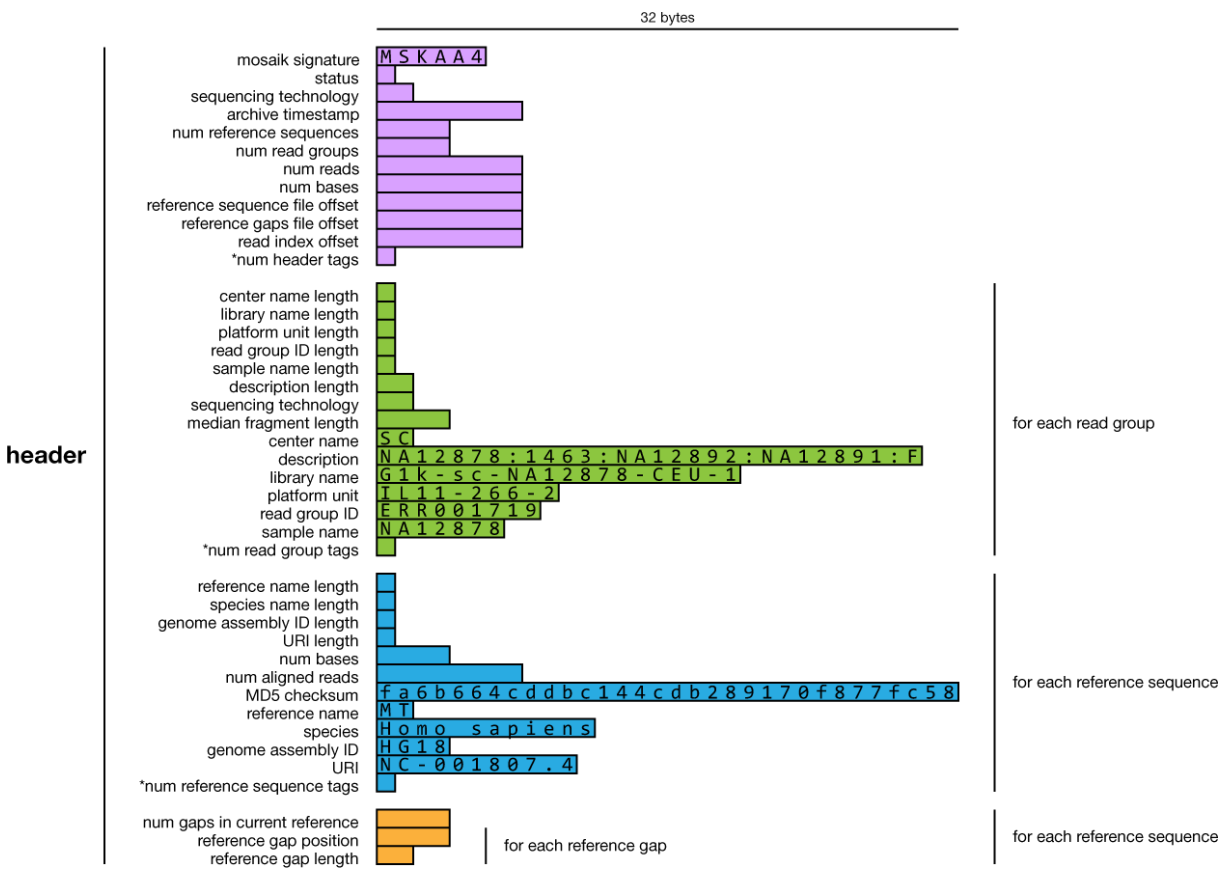

Figure S2. MOSAIK Alignment Archive Header. Statistics (total number of reads and bases) are kept in the header to facilitate quick retrieval. Status flags and sequencing technology enable downstream MOSAIK tools to remember how a data set was aligned. Read group records contain metadata related to the underlying sequences. Reference sequence records contain information related to how many reads aligned to a particular reference sequence in addition to reference name, species name, genome assembly ID, and the uniform resource identifier (URI).

When an insertion is noted in an alignment, the location and length of the insertion is noted. Recording these reference gaps help speed up the multiple sequence alignment creation in MosaikAssembler.

Tags are supported throughout the alignment archive in an attempt at making the alignment archive format resilient to additional fields that may be added in the future.

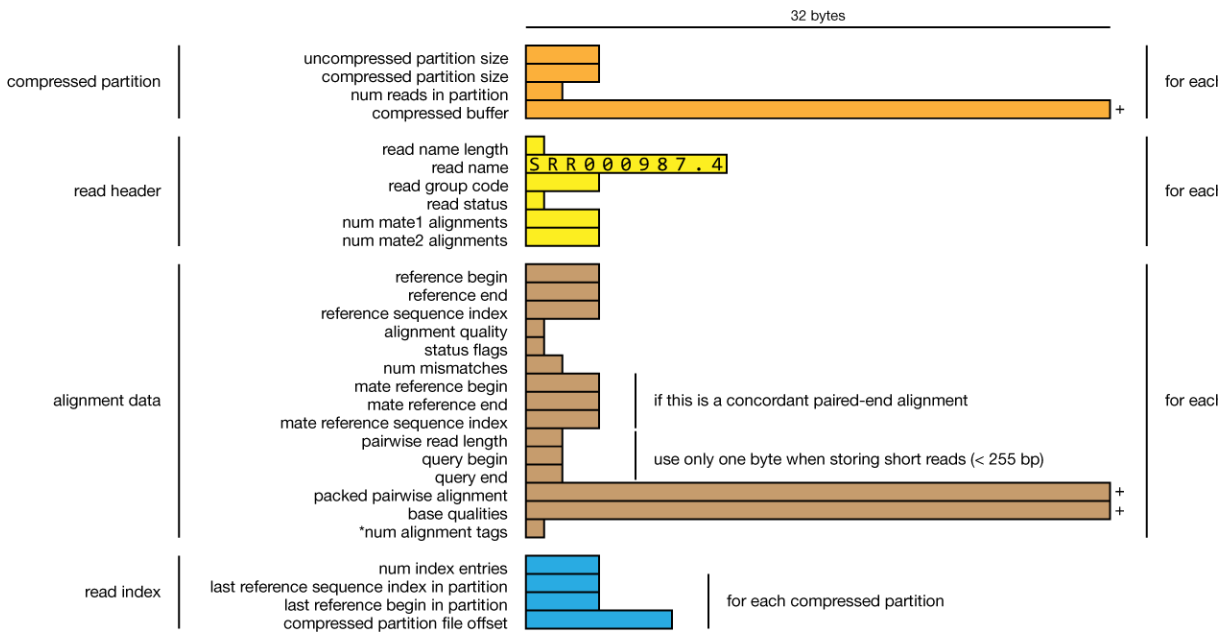

Figure S3. MOSAIK Alignment Archive Data. Each partition contains up to 20,000 aligned reads. When a partition fills up, the partition is compressed using the FastLZ algorithm. Each aligned read can contain any number of mate1 and mate2 alignments.

The packed pairwise alignment in the alignment data record converts the reference and query alignment into 4-bit notation. The reference alignment is then packed into the upper word of the query alignment. A 35 bp pairwise alignment will therefore only use 35 bytes instead of 70 bytes.

The read index is stored at the end of the alignment archive. In sorted files, it stores the last reference sequence index and last alignment start coordinate for each compressed partition. The index guarantees that the jump function will always find the closest point (within 20,000 alignments) before the desired alignment. Typical search times are on the order of half a second.

```
-------------------------------------------------------------------------------
AnalyzeSNPs                                                          2010-02-10
Michael Stromberg                 Marth Lab, Boston College Biology Department
-------------------------------------------------------------------------------

Initialization
- loading reference sequence... finished.
- loading Sanger SNPs...         338747 positions loaded.
- loading dbSNP SNPs...         1129748 positions loaded.
- loading Sanger INDELs...        24723 positions loaded.
- loading HapMap3 SNPs...         56071 positions loaded.
- loading GigaBayes variants... 409625 positions loaded.

Filtering
- merging consecutive SNPs... finished.
- merging consecutive INDELs... finished.
- calculating inter-SNP distance... finished.
- applying distance threshold to SNPs... finished.
- applying distance threshold to INDELs... finished.

Comparison
- comparing GigaBayes SNPs with dbSNP, Sanger, and HapMap3... finished.
- comparing GigaBayes INDELs with Sanger... finished.

Statistics
GigaBayes (unfiltered)
==================================================
# SNPs:                           361619
# INDELs:                          48006
# unknown:                             0

GigaBayes SNPs
==================================================
# Sanger SNPs in GigaBayes:      226505 ( 66.9 %)
# Sanger SNPs not in GigaBayes:  112242 ( 33.1 %)

# GigaBayes SNPs in dbSNP:       206373 ( 80.5 %)
# GigaBayes SNPs not in dbSNP:    49962 ( 19.5 %)

# dbSNP SNPs found:              206372 ( 18.3 %)
# dbSNP SNPs missed:             923376 ( 81.7 %)

# GigaBayes SNPs in HapMap:       53612 ( 20.9 %)
# GigaBayes SNPs not in HapMap:  202723 ( 79.1 %)

# HapMap SNPs found:              53612 ( 95.6 %)
# HapMap SNPs missed:              2459 (  4.4 %)

Sanger SNPs
==================================================
# GigaBayes SNPs in Sanger:      226505 ( 88.4 %)
# GigaBayes SNPs not in Sanger:   29830 ( 11.6 %)

# Sanger SNPs in dbSNP:          297074 ( 87.7 %)
# Sanger SNPs not in dbSNP:       41673 ( 12.3 %)

# dbSNP SNPs found by Sanger:    297074 ( 26.3 %)
# dbSNP SNPs missed by Sanger:   832674 ( 73.7 %)

# Sanger SNPs in HapMap:          55264 ( 16.3 %)
# Sanger SNPs not in HapMap:     283483 ( 83.7 %)

# HapMap SNPs found by Sanger:    55264 ( 98.6 %)
# HapMap SNPs missed by Sanger:     807 (  1.4 %)
```

```
Sanger INDELs
==================================================
# GigaBayes INDELs in Sanger:       19362 ( 58.2 %)
# GigaBayes INDELs not in Sanger:   13895 ( 41.8 %)
# Sanger INDELs not in GigaBayes:    5361 ( 21.7 %)

Miscellaneous
====================================================================
VENN: # SNPs in GigaBayes:                    23875 (  9.3 %)
VENN: # SNPs in Sanger:                       15586
VENN: # SNPs in dbSNP:                       826720
VENN: # SNPs in GigaBayes & Sanger:           26087 ( 10.2 %)
VENN: # SNPs in GigaBayes & dbSNP:             5955 (  2.3 %)
VENN: # SNPs in Sanger & dbSNP:               96656
VENN: # SNPs in GigaBayes & Sanger & dbSNP:  200418 ( 78.2 %)

VENN: # INDELs in GigaBayes:                  13895 ( 41.8 %)
VENN: # INDELs in Sanger:                      5361
VENN: # INDELs in GigaBayes & Sanger:         19362 ( 58.2 %)

# GigaBayes SNPs:                                 256335
# Sanger SNPs:                                    338747

# GigaBayes INDELs:                                33257
# Sanger INDELs:                                   24723

# GigaBayes transitions:                          162296
# GigaBayes transversions:                         91528
GigaBayes transition:transversion ratio:             1.8

GigaBayes SNP rate: 1 SNP per 877.8 bp
Sanger SNP rate:    1 SNP per 664.2 bp


AnalyzeSNPs wall time: 610.752 s
```

Figure S4. SNP analysis output for chromosome 1.

```
ALU.ALUSC
ALU.ALUSG
ALU.ALUSP
ALU.ALUSQ
ALU.ALUSX
ALU.ALUSZ
ALU.ALUY
ALU.ALUYA1
ALU.ALUYA4
ALU.ALUYA5
ALU.ALUYA8
ALU.ALUYB3A1
ALU.ALUYB3A2
ALU.ALUYB8
ALU.ALUYB9
ALU.ALUYBC3A
ALU.ALUYC1
ALU.ALUYC2
ALU.ALUYD2
ALU.ALUYD3
ALU.ALUYD3A1
ALU.ALUYD8
ALU.ALUYE2
ALU.ALUYE5
ALU.ALUYF1
ALU.ALUYF2
ALU.ALUYG6
ALU.ALUYH9
ALU.ALUYI6
L1.L1
L1.L1HS
L1.L1PA10
L1.L1PA11
L1.L1PA12
L1.L1PA12_5
L1.L1PA13
L1.L1PA13_5
L1.L1PA14
L1.L1PA14_5
L1.L1PA15
L1.L1PA16
L1.L1PA16_5
L1.L1PA17_5
L1.L1PA2
L1.L1PA3
L1.L1PA4
L1.L1PA5
L1.L1PA6
L1.L1PA7
L1.L1PA7_5
L1.L1PA8
SVA.SVA
```

Figure S5. The transposable element subclasses that were used for mobile element insertion discovery.

References

1. Wiesel, T. In celebration of the 50th anniversary of the publication of the experiment that transformed biology and showed that genes are made of DNA. (ed. Rockefeller University) 2 (New York, 1994).
2. Morgan, T.H. Sex Limited Inheritance in Drosophila. *Science* **32**, 120-122 (1910).
3. Avery, O.T., Macleod, C.M. & McCarty, M. Studies on the Chemical Nature of the Substance Inducing Transformation of Pneumococcal Types : Induction of Transformation by a Desoxyribonucleic Acid Fraction Isolated from Pneumococcus Type Iii. *J Exp Med* **79**, 137-158 (1944).
4. Franklin, R.E. & Gosling, R.G. Molecular configuration in sodium thymonucleate. *Nature* **171**, 740-1 (1953).
5. Watson, J.D. & Crick, F.H. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature* **171**, 737-8 (1953).
6. Wilkins, M.H., Stokes, A.R. & Wilson, H.R. Molecular structure of deoxypentose nucleic acids. *Nature* **171**, 738-40 (1953).
7. Crick, F.H. The biological replication of macromolecules. *Symp. Soc. Exp. Biol.* **12**, 138-163 (1958).
8. Crick, F.H., Barnett, L., Brenner, S. & Watts-Tobin, R.J. General nature of the genetic code for proteins. *Nature* **192**, 1227-32 (1961).
9. Nirenberg, M.W. & Matthaei, J.H. The dependence of cell-free protein synthesis in E. coli upon naturally occurring or synthetic polyribonucleotides. *Proc Natl Acad Sci U S A* **47**, 1588-602 (1961).
10. Blattner, F.R. et al. Charon phages: safer derivatives of bacteriophage lambda for DNA cloning. *Science* **196**, 161-9 (1977).
11. Mullis, K.B. & Faloona, F.A. Specific synthesis of DNA in vitro via a polymerase-catalyzed chain reaction. *Methods Enzymol* **155**, 335-50 (1987).
12. Saiki, R.K. et al. Diagnosis of sickle cell anemia and beta-thalassemia with enzymatically amplified DNA and nonradioactive allele-specific oligonucleotide probes. *N Engl J Med* **319**, 537-41 (1988).
13. Maxam, A.M. & Gilbert, W. A new method for sequencing DNA. *Proc Natl Acad Sci U S A* **74**, 560-4 (1977).
14. Sanger, F., Nicklen, S. & Coulson, A.R. DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci U S A* **74**, 5463-7 (1977).
15. Collins, F.S., Morgan, M. & Patrinos, A. The Human Genome Project: lessons from large-scale biology. *Science* **300**, 286-90 (2003).
16. Lander, E.S. et al. Initial sequencing and analysis of the human genome. *Nature* **409**, 860-921 (2001).
17. Finishing the euchromatic sequence of the human genome. *Nature* **431**, 931-45 (2004).
18. Venter, J.C. et al. The sequence of the human genome. *Science* **291**, 1304-51 (2001).
19. McPherson, J.D. et al. A physical map of the human genome. *Nature* **409**, 934-41 (2001).
20. Sanger, F. & Coulson, A.R. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J Mol Biol* **94**, 441-8 (1975).
21. Liolios, K. et al. The Genomes On Line Database (GOLD) in 2009: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Res* **38**, D346-54.
22. Clark, M.S. Comparative genomics: the key to understanding the Human Genome Project. *Bioessays* **21**, 121-30 (1999).
23. Mardis, E.R. Next-generation DNA sequencing methods. *Annu Rev Genomics Hum Genet* **9**, 387-402 (2008).
24. Drmanac, R. et al. Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. *Science* **327**, 78-81 (2010).
25. Margulies, M. et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* **437**, 376-80 (2005).
26. Shendure, J. & Ji, H. Next-generation DNA sequencing. *Nat Biotechnol* **26**, 1135-45 (2008).
27. Bentley, D.R. Whole-genome re-sequencing. *Current Opinion in Genetics & Development* **16**, 545-552 (2006).
28. Shendure, J. et al. Accurate multiplex polony sequencing of an evolved bacterial genome. *Science* **309**, 1728-32 (2005).

29. Applied Biosystems. Principles of Di-Base Sequencing and the Advantages of Color Space Analysis in the SOLiD System. 4 (Applied Biosystems, 2008).
30. Drmanac, R. et al. DNA sequence determination by hybridization: a strategy for efficient large-scale sequencing. *Science* **260**, 1649-52 (1993).
31. Pihlak, A. et al. Rapid genome sequencing with short universal tiling probes. *Nat Biotechnol* **26**, 676-84 (2008).
32. Harris, T.D. et al. Single-molecule DNA sequencing of a viral genome. *Science* **320**, 106-9 (2008).
33. Wade, N. Grad Student Becomes Gene Effort's Unlikely Hero. in *The New York Times* (New York, 2001). Retrieved from http://www.nytimes.com/2001/02/13/health/13HERO.html?pagewanted=1
34. Kent, W.J. & Haussler, D. Assembly of the Working Draft of the Human Genome with GigAssembler. *Genome Res* **11**, 1541-1548 (2001).
35. Myers, E.W. et al. A Whole-Genome Assembly of Drosophila. *Science* **287**, 2196-2204 (2000).
36. Huang, X., Wang, J., Aluru, S., Yang, S.P. & Hillier, L. PCAP: a whole-genome assembly program. *Genome Res* **13**, 2164-70 (2003).
37. Huang, X. & Madan, A. CAP3: A DNA sequence assembly program. *Genome Res* **9**, 868-77 (1999).
38. Batzoglou, S. et al. ARACHNE: a whole-genome shotgun assembler. *Genome Res* **12**, 177-89 (2002).
39. Green, P. phrap. Retrieved from http://www.phrap.org/phredphrapconsed.html
40. Pevzner, P.A., Tang, H. & Waterman, M.S. An Eulerian path approach to DNA fragment assembly. *Proc Natl Acad Sci U S A* **98**, 9748-53 (2001).
41. Zerbino, D.R. & Birney, E. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res* **18**, 821-9 (2008).
42. Jeck, W.R. et al. Extending assembly of short DNA sequences to handle error. *Bioinformatics* **23**, 2942-4 (2007).
43. Warren, R.L., Sutton, G.G., Jones, S.J. & Holt, R.A. Assembling millions of short DNA sequences using SSAKE. *Bioinformatics* **23**, 500-1 (2007).
44. Butler, J. et al. ALLPATHS: De novo assembly of whole-genome shotgun microreads. *Genome Res* **18**, 810-820 (2008).
45. Dohm, J.C., Lottaz, C., Borodina, T. & Himmelbauer, H. SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Res* **17**, 1697-1706 (2007).
46. Kent, W.J. BLAT--the BLAST-like alignment tool. *Genome Res* **12**, 656-64 (2002).
47. Homer, N., Merriman, B. & Nelson, S.F. BFAST: an alignment tool for large scale genome resequencing. *PLoS One* **4**, e7767 (2009).
48. Stromberg, M. MOSAIK: A next-generation reference-guided aligner. Retrieved from http://bioinformatics.bc.edu/marthlab/Mosaik
49. Li, H., Ruan, J. & Durbin, R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res* **18**, 1851-8 (2008).
50. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**, 1754-60 (2009).
51. Li, R. et al. SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* **25**, 1966-7 (2009).
52. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S.L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* **10**, R25 (2009).
53. Lam, T.W., Sung, W.K., Tam, S.L., Wong, C.K. & Yiu, S.M. Compressed indexing and local alignment of DNA. *Bioinformatics* **24**, 791-797 (2008).
54. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. Basic local alignment search tool. *J Mol Biol* **215**, 403-410 (1990).
55. The Broad Institute. The Genome Analysis Toolkit. Retrieved from http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit
56. Smith, T.F. & Waterman, M.S. Identification of common molecular subsequences. *J Mol Biol* **147**, 195-7 (1981).
57. Needleman, S.B. & Wunsch, C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* **48**, 443-453 (1970).
58. Li, H. et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078-2079 (2009).

59. Ning, Z., Caccamo, M. & Mullikin, J.C. ssahaSNP A Polymorphism Detection Tool on a Whole Genome Scale. in *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference - Workshops* 251-254 (IEEE Computer Society, 2005).
60. Marth, G.T. et al. A general approach to single-nucleotide polymorphism discovery. *Nat Genet* **23**, 452-6 (1999).
61. Li, R.Q. et al. SNP detection for massively parallel whole-genome resequencing. *Genome Res* **19**, 1124-1132 (2009).
62. Matukumalli, L.K. et al. Application of machine learning in SNP discovery. *Bmc Bioinformatics* **7**(2006).
63. Hoberman, R. et al. A probabilistic approach for SNP discovery in high-throughput human resequencing data. *Genome Res* **19**, 1542-52 (2009).
64. Shen, Y.F. et al. A SNP discovery method to assess variant allele probability from next-generation resequencing data. *Genome Res* **20**, 273-280 (2010).
65. Unneberg, P., Stromberg, M. & Sterky, F. SNP discovery using advanced algorithms and neural networks. *Bioinformatics* **21**, 2528-2530 (2005).
66. A haplotype map of the human genome. *Nature* **437**, 1299-320 (2005).
67. Cargill, M. et al. Characterization of single-nucleotide polymorphisms in coding regions of human genes. *Nat Genet* **22**, 231-8 (1999).
68. Halushka, M.K. et al. Patterns of single-nucleotide polymorphisms in candidate genes for blood-pressure homeostasis. *Nat Genet* **22**, 239-47 (1999).
69. Li, W.-H. *Molecular evolution*, xv, 487 p. (Sinauer Associates, Sunderland, Mass., 1997).
70. Feuk, L., Carson, A.R. & Scherer, S.W. Structural variation in the human genome. *Nat Rev Genet* **7**, 85-97 (2006).
71. Estivill, X. & Armengol, L. Copy Number Variants and Common Disorders: Filling the Gaps and Exploring Complexity in Genome-Wide Association Studies. *PLoS Genet* **3**, e190 (2007).
72. Chen, K. et al. BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat Meth* **6**, 677-681 (2009).
73. Stewart, D. SPANNER: a structural variation detection tool. Retrieved from http://bioinformatics.bc.edu/marthlab/Spanner
74. Ye, K., Schulz, M.H., Long, Q., Apweiler, R. & Ning, Z. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* **25**, 2865-2871 (2009).
75. Hormozdiari, F., Alkan, C., Eichler, E.E. & Sahinalp, S.C. Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Res* **19**, 1270-8 (2009).
76. Lee, S., Hormozdiari, F., Alkan, C. & Brudno, M. MoDIL: detecting small indels from clone-end sequencing with mixtures of distributions. *Nat Meth* **6**, 473-474 (2009).
77. Eid, J. et al. Real-time DNA sequencing from single polymerase molecules. *Science* **323**, 133-8 (2009).
78. Sherry, S.T. et al. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res* **29**, 308-311 (2001).
79. Adelson-Velskii, G. & Landis, E.M. An algorithm for the organization of information. *Proceedings of the USSR Academy of Sciences* **146**, 263–266 (1962).
80. Gotoh, O. An improved algorithm for matching biological sequences. *J Mol Biol* **162**, 705-8 (1982).
81. Shannon, C.E. A Mathematical Theory of Communication. *Bell System Technical Journal* **27**, 379-423 (1948).
82. Ewing, B. & Green, P. Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res* **8**, 186-94 (1998).
83. Bryson, A.E. & Ho, Y.-C. *Applied optimal control; optimization, estimation, and control*, 481 p. (Blaisdell Pub. Co., Waltham, Mass.,, 1969).
84. Wysoker, A., Tibbetts, K., Fennell, T. & Weisburd, B. Picard. Retrieved from http://picard.sourceforge.net/
85. Gordon, D. Viewing and editing assembled sequences using Consed. *Curr Protoc Bioinformatics* **Chapter 11**, Unit11 2 (2003).
86. Marth, G. (2010-02-14) GigaBayes. Retrieved from http://bioinformatics.bc.edu/marthlab/GigaBayes
87. Free Software Foundation, I. (2010-04-09) GNU General Public License v2.0. Retrieved from http://www.gnu.org/licenses/old-licenses/gpl-2.0.html

88. Hidayat, A. FastLZ - free, open-source, portable real-time compression library. Retrieved from http://www.fastlz.org/
89. Hercus, C. (2010-03-14) Novoalign. Retrieved from http://www.novocraft.com/
90. Bentley, D.R. et al. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* **456**, 53-9 (2008).
91. Anderson, P. & Li, Y. (2010-04-11) KARMA. Retrieved from http://www.sph.umich.edu/csg/pha/karma/index.html
92. Burrows, M. & Wheeler, D.J. A Block-sorting Lossless Data Compression Algorithm. (Digital Equipment Corporation, Palo Alto, California, 1994).
93. Burkhard, W.A. & Keller, R.M. Some approaches to best-match file searching. *Commun. ACM* **16**, 230-236 (1973).
94. Genome sequence of the nematode C. elegans: a platform for investigating biology. *Science* **282**, 2012-8 (1998).
95. Waterston, R. et al. The genome of the nematode Caenorhabditis elegans. *Cold Spring Harb Symp Quant Biol* **58**, 367-76 (1993).
96. Harris, T.W. et al. WormBase: a multi-species resource for nematode biology and genomics. *Nucleic Acids Res* **32**, D411-7 (2004).
97. Stein, L.D. et al. The genome sequence of Caenorhabditis briggsae: a platform for comparative genomics. *PLoS Biol* **1**, E45 (2003).
98. Hodgkin, J. & Doniach, T. Natural variation and copulatory plug formation in Caenorhabditis elegans. *Genetics* **146**, 149-64 (1997).
99. The ENCODE (ENCyclopedia Of DNA Elements) Project. *Science* **306**, 636-40 (2004).
100. Gordon, P. XML for Molecular Biology as compiled by Paul Gordon. Retrieved from http://www.visualgenomics.ca/gordonp/xml/
101. Smit, A., Hubley, R. & Green, P. RepeatMasker Open-3.0. Retrieved from www.repeatmasker.org
102. Morgulis, A., Gertz, E.M., Schaffer, A.A. & Agarwala, R. A fast and symmetric DUST implementation to mask low-complexity DNA sequences. *J Comput Biol* **13**, 1028-40 (2006).
103. Malde, K., Schneeberger, K., Coward, E. & Jonassen, I. RBR: library-less repeat detection for ESTs. *Bioinformatics* **22**, 2232-6 (2006).
104. Stephens, M., Sloan, J.S., Robertson, P.D., Scheet, P. & Nickerson, D.A. Automating sequence-based detection and genotyping of SNPs from diploid samples. *Nat Genet* **38**, 375-381 (2006).
105. Suh, S.O., Marshall, C.J., McHugh, J.V. & Blackwell, M. Wood ingestion by passalid beetles in the presence of xylose-fermenting gut yeasts. *Mol Ecol* **12**, 3137-45 (2003).
106. Parekh, S.R., Parekh, R.S. & Wayman, M. Fermentation of Xylose and Cellobiose by Pichia-Stipitis and Brettanomyces-Clausenii. *Applied Biochemistry and Biotechnology* **18**, 325-338 (1988).
107. Jeffries, T.W. et al. Genome sequence of the lignocellulose-bioconverting and xylose-fermenting yeast Pichia stipitis. *Nat Biotechnol* **25**, 319-26 (2007).
108. Applied Biosystems. (2010-03-25) SOLiD system application documentation: AB resequencing analysis pipeline (Corona Lite). Retrieved from http://solidsoftwaretools.com/gf/project/corona/
109. Richly, E. & Leister, D. NUMTs in sequenced eukaryotic genomes. *Mol Biol Evol* **21**, 1081-4 (2004).
110. Bensasson, D., Zhang, D., Hartl, D.L. & Hewitt, G.M. Mitochondrial pseudogenes: evolution's misplaced witnesses. *Trends Ecol Evol* **16**, 314-321 (2001).
111. Lopez, J.V., Yuhki, N., Masuda, R., Modi, W. & O'Brien, S.J. Numt, a recent transfer and tandem amplification of mitochondrial DNA to the nuclear genome of the domestic cat. *J Mol Evol* **39**, 174-90 (1994).
112. Gropp, W., Lusk, E., Doss, N. & Skjellum, A. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing* **22**, 789-828 (1996).
113. Olson, M.A., Bostic, K. & Seltzer, M.I. Berkeley DB. in *USENIX Annual Technical Conference, FREENIX Track* 183-191 (USENIX, Monterey, California, 1999).
114. 1000 Genomes Project Consortium. (March 11, 2010) 1000 Genomes - About. Retrieved from http://www.1000genomes.org/page.php?page=about
115. Coriell Institute for Medical Research. (March 11, 2010) GM12878. Retrieved from http://ccr.coriell.org/Sections/Search/Sample_Detail.aspx?Ref=GM12878&PgId=166

116. Sequenom. (March 11, 2010) iPlex for SNP Genotyping. Retrieved from http://www.sequenom.com/Genetic-Analysis/Applications/iPLEX-Genotyping/iPLEX_How-it-Works
117. Cordaux, R. & Batzer, M.A. The impact of retrotransposons on human genome evolution. *Nat Rev Genet* **10**, 691-703 (2009).
118. Mills, R.E., Bennett, E.A., Iskow, R.C. & Devine, S.E. Which transposable elements are active in the human genome? **23**, 183-191 (2007).
119. Coufal, N.G. et al. L1 retrotransposition in human neural progenitor cells. *Nature* **460**, 1127-31 (2009).
120. Faulkner, G.J. et al. The regulated retrotransposon transcriptome of mammalian cells. *Nat Genet* **41**, 563-71 (2009).
121. Belancio, V.P., Deininger, P.L. & Roy-Engel, A.M. LINE dancing in the human genome: transposable elements and disease. *Genome Med* **1**, 97 (2009).
122. Lev-Maor, G. et al. Intronic Alus influence alternative splicing. *PLoS Genet* **4**, e1000204 (2008).
123. Levy, S. et al. The diploid genome sequence of an individual human. *PLoS Biol* **5**, e254 (2007).
124. Xing, J. et al. Mobile elements create structural variation: analysis of a complete human genome. *Genome Res* **19**, 1516-26 (2009).
125. Ahn, S.M. et al. The first Korean genome sequence and analysis: full genome sequencing for a socio-ethnic group. *Genome Res* **19**, 1622-9 (2009).
126. Kim, J.I. et al. A highly annotated whole-genome sequence of a Korean individual. *Nature* **460**, 1011-5 (2009).
127. Wang, J. et al. The diploid genome sequence of an Asian individual. *Nature* **456**, 60-5 (2008).
128. Wheeler, D.A. et al. The complete genome of an individual by massively parallel DNA sequencing. *Nature* **452**, 872-6 (2008).
129. National Center for Biotechnology Information. (March 15, 2010) SRA Home. Retrieved from http://www.ncbi.nlm.nih.gov/sra
130. Jurka, J. et al. Repbase Update, a database of eukaryotic repetitive elements. *Cytogenet Genome Res* **110**, 462-7 (2005).
131. Wang, J. et al. dbRIP: a highly integrated database of retrotransposon insertion polymorphisms in humans. *Hum Mutat* **27**, 323-9 (2006).
132. Kidd, J.M. et al. Mapping and sequencing of structural variation from eight human genomes. *Nature* **453**, 56-64 (2008).
133. Mills, R.E. et al. Recently mobilized transposons in the human and chimpanzee genomes. *Am J Hum Genet* **78**, 671-9 (2006).
134. Initial sequence of the chimpanzee genome and comparison with the human genome. *Nature* **437**, 69-87 (2005).
135. Wang, H. et al. SVA Elements: A Hominid-specific Retroposon Family. *J Mol Biol* **354**, 994-1007 (2005).
136. Ostertag, E.M., Goodier, J.L., Zhang, Y. & Kazazian Jr, H.H. SVA Elements Are Nonautonomous Retrotransposons that Cause Disease in Humans. *The American Journal of Human Genetics* **73**, 1444-1451 (2003).
137. Gu, Y. et al. The first reported case of Menkes disease caused by an Alu insertion mutation. *Brain and Development* **29**, 105-108 (2007).
138. Kazazian, H.H. Mobile elements and disease. *Curr Opin Genet Dev* **8**, 343-350 (1998).
139. Kazazian, H.H., Jr. Mobile Elements: Drivers of Genome Evolution. *Science* **303**, 1626-1632 (2004).
140. Harrow, J. et al. GENCODE: producing a reference annotation for ENCODE. *Genome Biol* **7**, S4 (2006).
141. Zhu, L. et al. Patterns of exon-intron architecture variation of genes in eukaryotic genomes. *BMC Genomics* **10**, 47 (2009).
142. Rasmussen, M. et al. Ancient human genome sequence of an extinct Palaeo-Eskimo. *Nature* **463**, 757-762 (2010).
143. Green, R.E. et al. Analysis of one million base pairs of Neanderthal DNA. *Nature* **444**, 330-336 (2006).
144. Harris, M. Mapping computational concepts to GPUs. in *ACM SIGGRAPH 2005 Courses* 50 (ACM, Los Angeles, California, 2005).
145. Boss, G., Malladi, P., Quan, D., Legregni, L. & Hall, H. Cloud Computing. 17 (IBM Corporation, 2007).

146. Schatz, M., Trapnell, C., Delcher, A. & Varshney, A. High-throughput sequence alignment using Graphics Processing Units. *Bmc Bioinformatics* **8**, 474 (2007).
147. Amazon Web Services LLC. (2010-04-09) Amazon Elastic Compute Cloud (Amazon EC2). Retrieved from http://aws.amazon.com/ec2/
148. Amazon Web Services LLC. (2010-04-09) Amazon Simple Storage Service (Amazon S3). Retrieved from http://aws.amazon.com/s3/
149. The Apache Software Foundation. (2010-04-09) Welcome to Apache Hadoop! Retrieved from http://hadoop.apache.org/
150. Dean, J. & Ghemawat, S. MapReduce: Simplified Data Processing on Large Clusters. 137-150.
151. Schatz, M.C. CloudBurst: highly sensitive read mapping with MapReduce. *Bioinformatics* **25**, 1363-1369 (2009).
152. Cock, P.J.A. et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422-1423 (2009).
153. The Apache Software Foundation. (2010-04-09) Welcome to Hadoop Distributed File System! Retrieved from http://hadoop.apache.org/hdfs/
154. Flynn, M. Some Computer Organizations and Their Effectiveness. *IEEE Trans. Comput.* **C-21**, 948 (1972).
155. Duncan, R. A Survey of Parallel Computer Architectures. *Computer* **23**, 5-16 (1990).
156. Rognes, T. & Seeberg, E. Six-fold speed-up of Smith-Waterman sequence database searches using parallel processing on common microprocessors. *Bioinformatics* **16**, 699-706 (2000).
157. Farrar, M. Striped Smith-Waterman speeds database searches six times over other SIMD implementations. *Bioinformatics* **23**, 156-161 (2007).