Developing computational strategies for assembly of heterozygous DNA sequence data

Su Sarlar susarlar@sabanciuniv.edu
Managerial Sciences, Senior

Samuel Lee samuellee@sabanciuniv.edu
Computer Science & Engineering, Sophomore

Stuart J Lucas
Plant Molecular Biology/Bioinformatics

Abstract

The process of genome assembly, recently possible due to technological advancements, has presented large amounts of genetic sequence data. However, DNA must first be split into countless small fragments which must be read, compared and merged to recover the original genome sequence. Specifically focusing on *Corylus avellana*, also known as the hazelnut, the focus of the project is to work with a large whole-genome sequence dataset, a diploid genome with high heterozygosity, to determine the original genome sequence. Multiple existing programs and software was used to develop new solutions to solve issues of highly heterozygous genomes and create a more integrated and holistic genome assembly.

Keywords: DNA, *de novo* Genome Assembly, heterozygosity, haplotype, *in silico*

1 Introduction

DNA sequencing technology has been rapidly developing in the recent years with fast-pace technical improvements ever since the International Human Genome Sequencing Consortium published the first draft of the human genome in 2001. This first draft of the human genome consisted of three billion base pairs, and a full sequence was completed and published in 2003(International Human Genome Sequencing Consortium, 2016). Since then, technological advancements have made it possible to obtain large amounts of genetic sequence data for almost any biological organism, including species of plants, animals and other organisms. Most of the current sequencing technologies share the same method of retrieving the genome, which splits the cellular DNA into thousands, or if needed millions of small sequence fragments or "reads."

The reconstruction of the original genome sequence from these reads is called 'sequence assembly'. The construction of reads from raw data is error prone, and the rate of error has a tendency to be based technology. Problems may be due to errors in raw read data due to platform specific chemistry, imaging and PCR incorporation errors(Harismendy et al, 2009) . For example

the Nanopore data has a high rate of error but provides longer reads. Pacbio also provides long reads, and the rate of errors is less than that of Nanopore. Illumina sequencing data gives shorter but more accurate reads, with higher coverage; that is a Illumina sequencing data can result in some long assembled sequences, however the sequences are not long enough to complete the whole genome: after the assembly is done there are places where can not be connected and these disruptions are called gaps. Usually Illumina coupled with PacBio or Nanopore reads are preferred for a reliable genome assembly, as was used in this project.

Another source of error is most specialized programs assuming that the DNA sequence is haploid, having only a single copy of each haplotype. This may present a serious problem as many eukaryotic organisms are either diploid or polypoid, having multiple copies of each chromosome. All non-identical and differing copies of each gene are crucial to understanding specific biological traits of the organism, including the genetic basis of disease, inheritance of traits and more.

The purpose of this project was to work with a specific heterozygous diploid genome to develop a more holistic and realistic genome assembly. Previously, initial genome assembly of *Corylus avellana*, also known as European hazelnut, produced large numbers of duplicated elements and a larger than expected genome size, which implies problems occurring due to heterozygosity. Working on the large whole-genome sequence data and using various existing tools and different software programs, the goal of the project was to solve issues of heterozygosity and develop new ways to filter and present data for a more complete genome assembly. Through numerous data filtering and analysis procedures, by the end of the project, a data table was created to show the different areas of heterozygosity, with the information of the nucleotide, type of mapping (insertion, deletion or single nucleotide polymorphism(SNP)) and starting and ending position of the heterozygous section, each matched with a specific consensus ID. Using specific conditions during the filtering process, including depth and quality, the first final data table was smaller than expected, which allows for further research and testing in different filtering processes and conditions to obtain a more realistic genome sequence.

2 Methods and Results

2.1 Platanus-allee

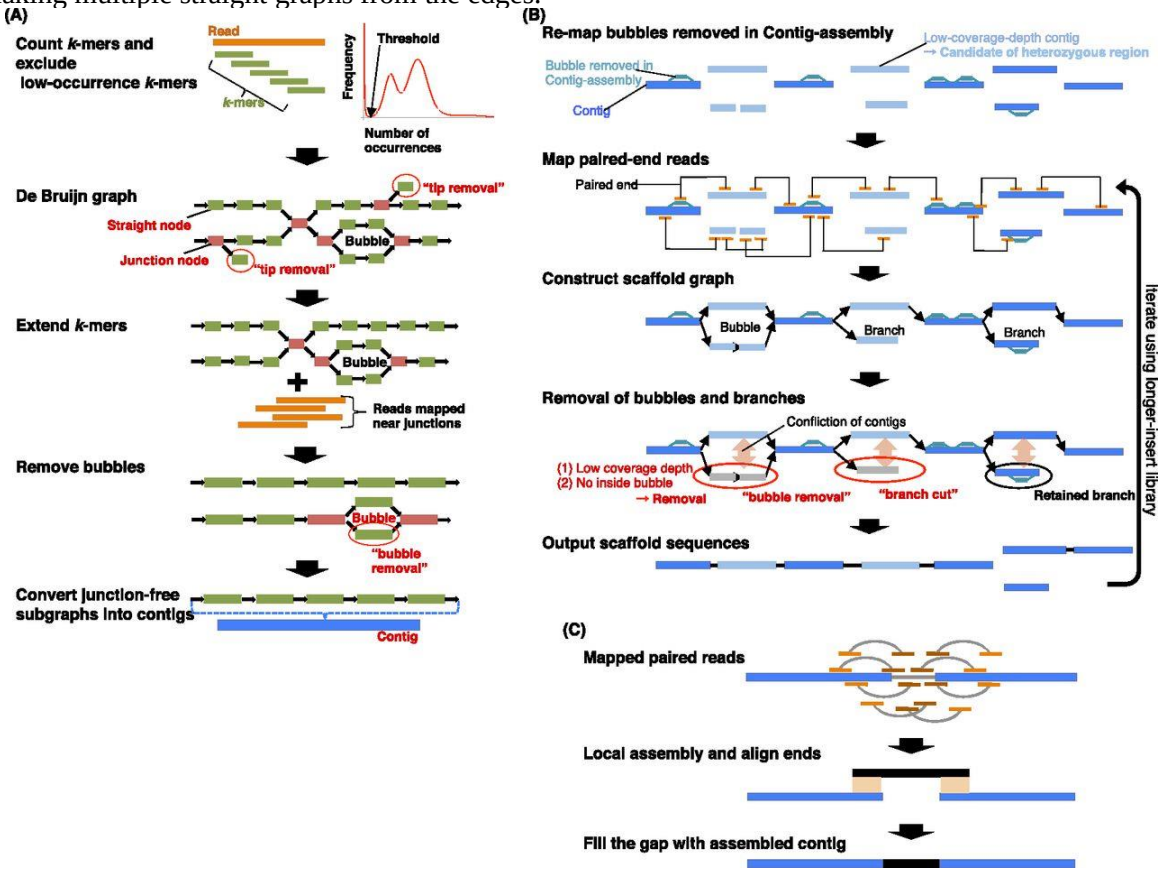
Platanus-allee is a *de novo* sequence assembler for diploid genomes (http://platanus.bio.titech.ac.jp/platanus2). It may be preferred for analysis of highly heterozygous genomes from massively parallel shotgun sequencing (next generation sequencing) data compared to other assemblers. In highly heterozygous regions, haplotypes extremely differ. Other assemblers give one merged sequence by eliminating one of the different sequences; Platanus allee however assembles two haplotypes separately and gives 2 sequences instead of one, so it is a haplotype aware genome assembly.

In order to assemble our haplotypes, find out homozygous regions where the sequences are likely to be assembled into a continuous string, and heterozygous regions where there are multiple ways that a continuous string can be formed, we have used Platanus-allee, with Nanopore reads and Illumina reads as inputs to Platanus-allee. The term for continuous strings, a set of overlapping DNA pieces representing a consensus region of DNA is “contig” and the term for DNA piece is “read”.

The algorithm of Platanus-allee uses De Bruijn graphs to assemble reads into contigs and uses optimized kmers in this process; nodes represent kmers and edges represent k-1 overlaps between

kmers. Differently than prior assemblers, Platanus automatically extends kmers to handle big and repetitive data. Once contigs are formed, they are scaffolded based on paired end libraries or mate pair libraries. In these contig assembly and scaffolding steps, complicated graph structures are simplified. Contig and scaffold construction is based on graphs without junctions; that is if a node has multiple edges.

One obstacle of application of de Bruijn graph based assembly is heterozygosity between diploid chromosomes. When the organism is diploid, heterozygous regions will have different kmers, and this will lead to junctions: the transition points between homozygous and heterozygous parts. The presence of differences will lead to bubbles. Bubbles are dealt by removing edges from junctions and making multiple straight graphs from the edges.



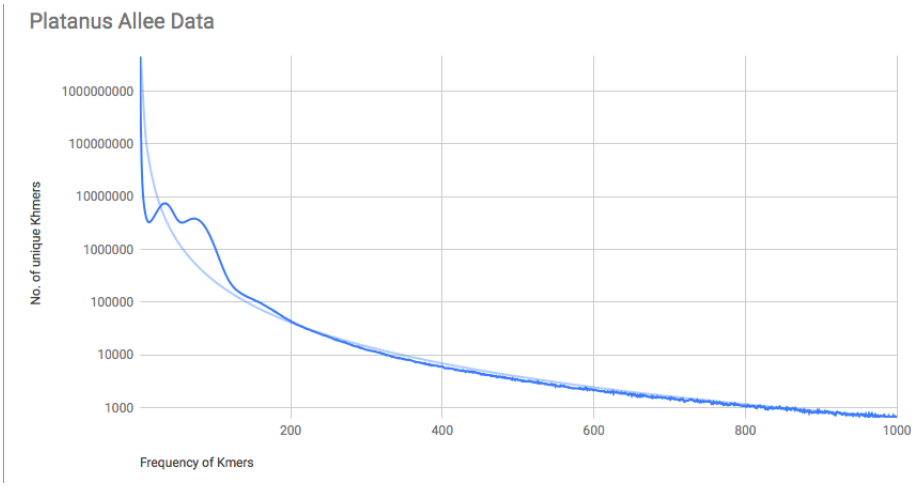
(Overview of the Platanus algorithm, from Kajitani et al, 2014)

(A) Construction of de Bruijn graph from data; there are some short branches, which are caused by errors. Short branches are removed via tip removal. Via kmer extension that maps previous graphs and reads to kmers at junctions, short repeats are resolved. Bubble structures that were caused by heterozygosity or errors are removed in the last step, and we obtain contigs.

(B)Scaffolding step includes detection of links between contigs using paired reads. In Contig assembly step the bubbles are removed, remapped on contigs and used for mapping of paired-end reads and to detect heterozygous contigs. With bubble removal or branch cut step the heterozygous regions are removed as bubble or branch structures.

(C) Gap-close step includes, mapping of paired reads on scaffolds, and mapping of reads at nearby gaps. If a contig is constructed from collected reads and may be assumed to cover the gap, the gap is closed by the contig.

2.2 Platanus-Allee Output



Platanus-Allee Data graph

With the data set of the Platanus-allee, the result was inserted into a graph using a logarithmic scale. The x-axis represented the frequency of the kmers, while the y-axis represented the number of unique kmers. The purpose of the graph was to determine areas of heterozygosity in the output by comparing areas of overlap in frequency of the kmers against the number of unique kmers. As seen in the graph above, the first left peak is the area of heterozygosity, as the number of unique kmers are shown to rise again as frequency of the kmers increase. The places of overlapping frequency for each unique number of kmer is the areas of heterozygosity. In addition, the part of the graph, as it approaches 200 in frequency value, is questionable whether it is heterozygous or not, as it shows possible points of heterozygosity. However, the graph reveals clearly that heterozygous regions exist within the Platanus-allee data set.

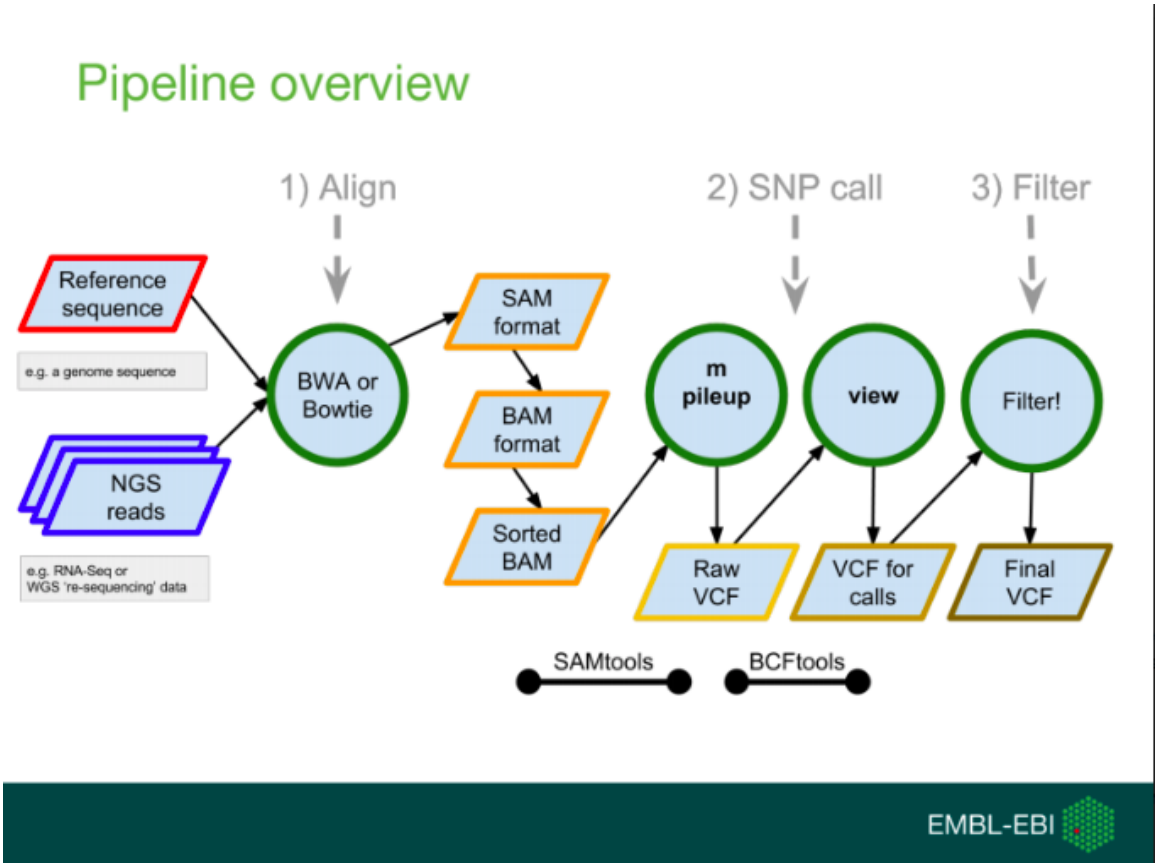
2.3 File production

Throughout the project, there were multiple filtering processes to determine the heterozygous area of the genome. The first step included using “bwa- Burrows-Wheeler Alignment Tool,” a software package that consists of different algorithms to map low-divergent sequences with a large reference genome(Heng, 2010). To successfully carry out this step in this project, a consensus scaffold fasta file was selected, which was to be indexed. For the algorithm to function efficiently, the BWA constructs indexes for the reference genome(Li & Durbin, 2009). Although there were several sub-commands to determine different alignment outputs, the most basic command “index” was sufficient for the reference fasta file. Afterwards, a “bwa mem” command was used to use the BWA-MEM algorithm which initially seeds alignments with maximal exact matches, then extends the seeds with the affine-gap Smith-Waterman algorithm. The option “-t” was used to specifically determine the number of threads for parallel computing.

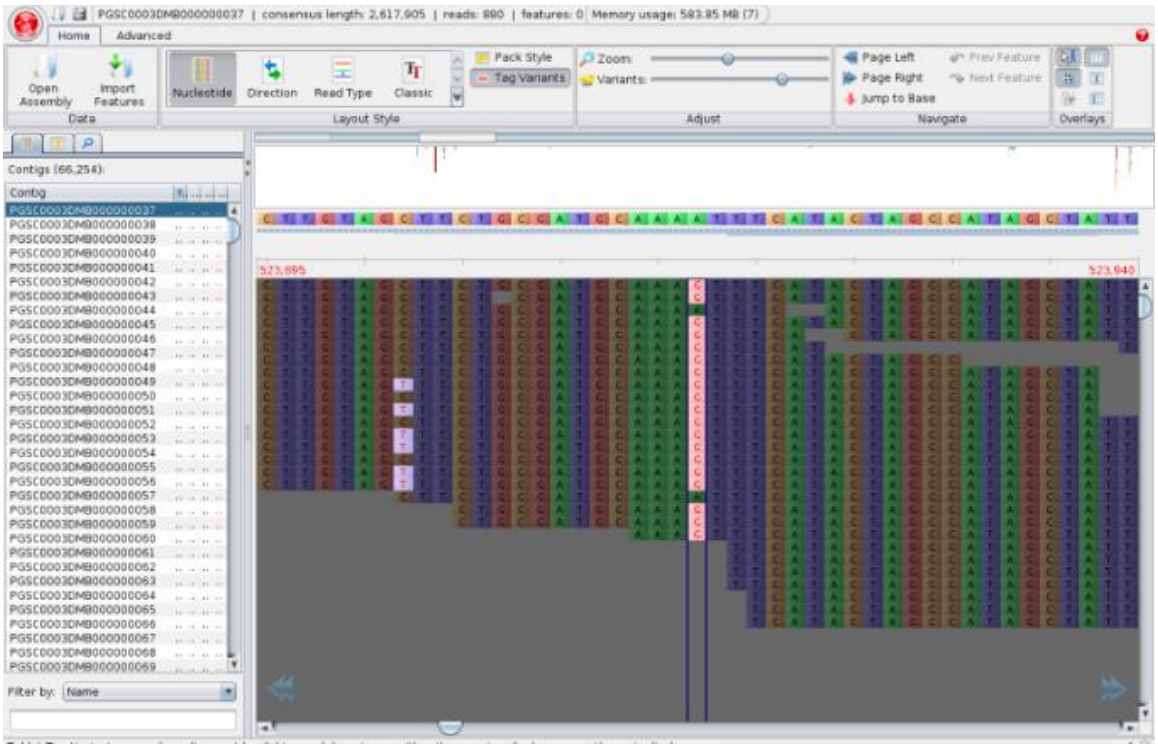
This created a .SAM output file, which was further processed and converted to a Binary Alignment/Map (BAM) file using samtools. Samtools uses the Sequence Alignment/Map file format to sort, merge, index and retrieve reads in any region, and is able to import or export files in both SAM and BAM format(Center for Statscal Analysis, 2010). Once the sorted BAM file

was created, bcftools – a tool for Binary Call Format (BCF) and VCF – was finally used to create a VCF file. The BCF file was generated using the "mpileup" command which revealed information about the match, mismatch, indel, strand, mapping quality and start and end of a read, separated by each line which represented a genomic position with chromosome name, coordinate, reference base, read bases, read qualities and alignment mapping qualities(Li, 2009). Then with the raw VCF, although the "view" command was used to extract all the sub-alignments, including areas of overlapping regions, there was an updated version of bcftools module, which allowed us to call a consensus and output only the variant sites. Furthermore, specific conditions could be applied to control the filtered VCF file, including depth and quality. Throughout this process, another software application called "Tablet" was used to help give further understanding, providing a more visual representation of the sequence alignment map. Tablet is a high-performance graphical view software for visualization of assemblies and read mappings. It revealed the depth of the genome sequence and the areas of overlap and places of potential insertion/deletion to determine the heterozygous parts of the sequence genome(Milne et al, 2013).

With the final vcf file, further extraction was done to ensure that the scaffold ends were handled correctly by using the "query" command in bcftools. Using this command, an table output file consisting of specific information that was requested was made, including the consensus ID of the scaffold, the position and ending position in the sequence and type of mapping (insertion/deletion). This final step was taken to analyze the data and to determine the starting and ending position of the heterozygous section.



Overview of file production



Tablet Sample

2.4 Further data filtering (excel)

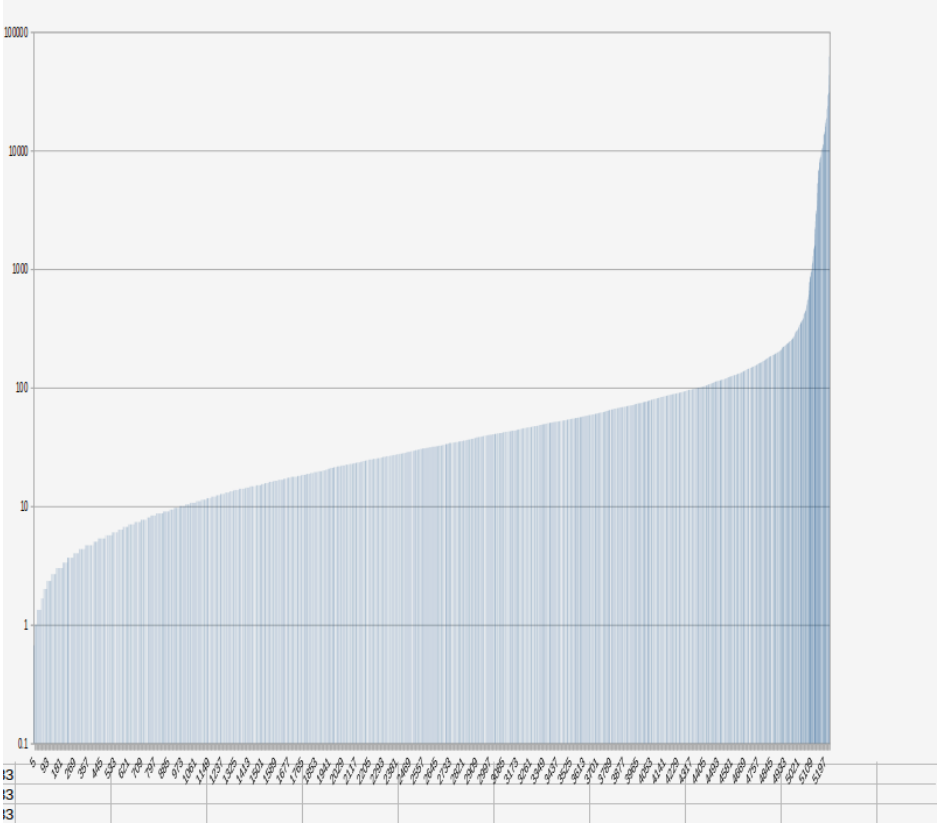
scaffold name	start pos	end pos
scaffold1057_len154451_cov42.4636_read151_maxK101	133895	136591
scaffold1331_len69105_cov45.0522_read151_maxK101	44465	64515
scaffold14774_len15265_cov44.5163_read151_maxK101	5941	8837
scaffold1509_len131609_cov44.4858_read151_maxK101	27577	68167
scaffold1509_len131609_cov44.4858_read151_maxK101	68167	82042
scaffold156_len124850_cov42.4795_read151_maxK101	30405	31494

Beginning of final data table indicating heterozygous regions

This table was constructed by filtering the final vcf file with a Phred quality score of 30 (indicating an 1 in each 1000 bases may be erroneous) and a depth of 2 (each sequence should be covered at least twice). The necessary columns were extracted using the command:

```
> bcftools query -f "%CHROM\t%POS\t%TYPE\n"
   Filtered_tombul_filteredq30d2.vcf > table_q30d2filtered3.tbl
```

As we had 6 375 variant sites in our filtered vcf file, we divided the genome size (380 000 000) by this number and found an average of about 59600; indicating if everything is random, then there should be one variation per 59600 bases. However assuming that there is heterozygosity, some regions should have more condensed SNVs. To find the condensed regions we decided to use a window frame of 3 to find a score for position of each SNV within each contig. If the name of contig before and after were the same, it would print the difference between previous and next SNV position divided by 3 (score), if the name of the contig is not the same, it would print "nd" as the number is irrelevant. As there is a jump of data after 320 (log320=2.5), we used 320 to compare the window score. If score was less than 320 it was marked as 1 and if not 0.
Next, if a position was marked as 1 and had a 0 before it (region not heterozygous, score>320) it was marked as a heterozygous region start. If the position was marked as 1 and the position after it was marked as 0, it was marked as heterozygous region end. In the final table, scaffold names, starting and ending positions of heterozygous regions were noted.



Occurrences of window scores (y axis score, x axis number of occurrence except nd)

3 Conclusion and Future Work

The results of this project may be questionable. A big proportion of the SNV's given the final vcf data ranked a Phred score of about 25, indicating that about one in 316 SNV reportings may actually be an error. The criteria we have used was to include SNP's with a Phred score above 30: only one in every 1000 reported SNV's would be an error. It is likely that this criteria was too stringent and excluded too much data. The vcf filtering step may be repeated to include SNV's with a Phred score more than 24. In the original Platanus paper (Kajitani 2014) it was noted a coverage larger than 100 was optimal, however in this project we have used a depth of at least 2, as larger coverage criteria might have eliminated a larger set of data. The accuracy of the last statement is dependent on how much did Platanus allee improve compared to Platanus. There are newer softwares being developed and published: such as ones promising to work with lower coverage, but unable to handle repetitive regions yet, or previously released softwares being improved such as Meraculous-2D (Goltsman 2017). Meraculous 2 D claims to give superior results to Platanus.

Some of the steps of this project may be dependent on experience and require deeper knowledge than the students had, especially those with seemingly intuitive steps of setting a window size or selecting quality and coverage criteria.

Our results were necessary for proceeding to next steps of genome assembly of Tombul cultivar of hazelnut. Currently we know what regions are heterozygous but we do not know what haplotypes have got which heterozygous regions. Perhaps other software such as Meraculous-2D can be used.

References

An Overview of Human Genome Project. (2016, May 11). National Human Genome Research Institute. Retrieved from https://www.genome.gov/12011238/an-overview-of-the-human-genome-project/.

Center for Statistical Analysis. (2013, 26 February). BAM. Retrieved from https://genome.sph.umich.edu/wiki/BAM

Garg, S., Rautiainen, M., Novak, A.M., Garrison, E., Durbin, R., & Marschall, T. (2018). A graph-based approach to diploid genome assembly. Bioinformatics. Volume 34, Issue 13, 1 July 2018, Pages i105–i114, https://doi.org/10.1093/bioinformatics/bty279

Goltsman, E., Ho, I.Y., & Rokhsar, D. (2017). Meraculous-2D: Haplotype-sensitive Assembly of Highly Heterozygous genomes.

Harismendy, O., Ng, P., Strausberg, R., Wang, X., Stockwell, T., Beeson, K., Schork, N., Murray, S., Topol, E., Levy, S., Frazer, K. Evaluation of next generation sequencing platforms for population targeted sequencing studies. Genome biol. 2009;10:R32

Heng, L. (2011, July 5). Manual Reference Pages - samtools (1). Retrieved from http://samtools.sourceforge.net

Heng, L. (2010, February 28). Burrows-Wheeler Aligner. Retrieved from http://bio-bwa.sourceforge.net

Heng, L. Bcftools. Retrieved from http://www.htslib.org/doc/bcftools.html

Kajitani R, Toshimoto K, Noguchi H, Toyoda A, Ogura Y, Okuno M, Yabana M, Harada M, Nagayasu E, Maruyama H, Kohara Y, Fujiyama A, Hayashi T, Itoh T, "Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads". Genome Res. 2014 Aug;24(8):1384-95. doi: 10.1101/gr.170720.113.

Li & Durbin 2009, Bioinformatics 14:1754-60)

Milne, I., Stephen, G., Bayer, M., Cock, P., Pritchard, L. Cardle, L., Shaw, P., Marshall, D. (2013). Using Tablet for visual exploration of second-generation sequencing data, Briefings in Bioinformatics, Volume 14(2), pp.193-202.