**Whole Genome Sequencing Primer**

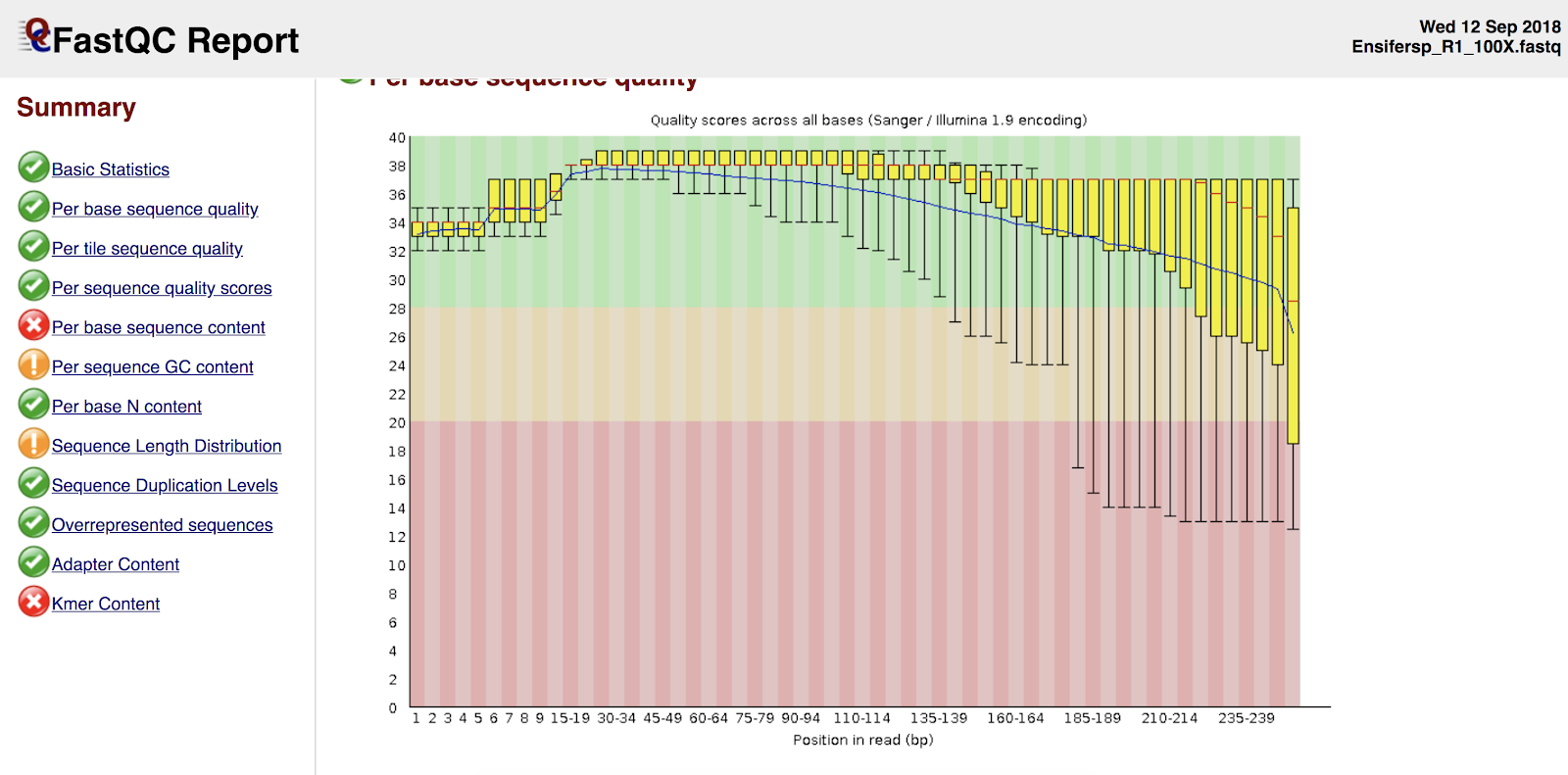
Lyndi Earnshaw & Andrea Soles

**Introduction:**

Being able to sequence and break apart a whole genome sequence is something very important when it comes to genetics and other aspects of biology. By being able to sequence the genes within different organisms it allows for scientists to better understand the organism, this overall process is used to determine the entirety of a DNA sequence of a genome at once. The Linux program is one of the ways a genome can be sequenced. Linux is an operating system that controls communication between the software and hardware on one’s computer (Linux.com, 2016). Specifically, for genome sequencing, a Linux program, Putty, is used. Putty allows remote access to another computer, this allows for people to load data from their computer and send it to a server that can decipher the information (Linux.com, 2016). This paper will be able to walk an undergraduate or graduate student through next generation sequencing methods targeted at whole genome sequencing of an unknown organism.

**Genome Assembly- Quality Control:**

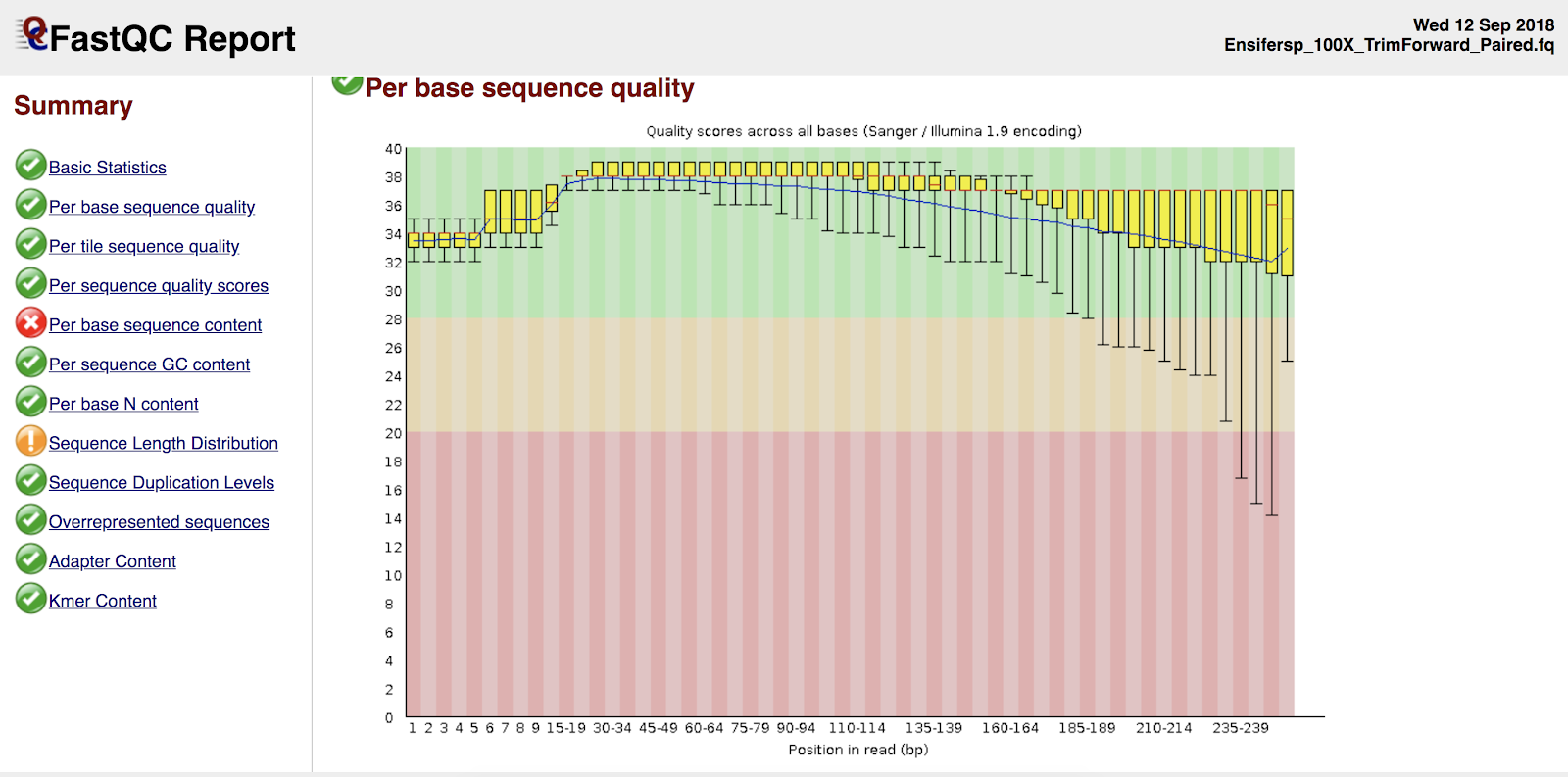
    The first thing that is needed to start sequencing is raw data that comes from the organism that is trying to be sequenced. The quality of the sequence needs to be high, this can be checked using *Fastqc.* This program allows the data to be read and checked for any errors that are within the sequence, it will generate a QC report to show and identify the errors that are in the raw data sequence in the starting library (Andrews, 2010). The *Fastqc* output generates four different lines of data, the first being the forward read of the DNA fragment, the second line is the actual sequence, the third line separates the sequence and the quality scores of the overall sequence, and the last line is the Phred scores that indicates the quality of the genomic DNA sequence (Wikipedia: FASTQ, 2018).To be able to see the *Fastqc* results a program called *Cyberduck* is used to see the graphs and charts displaying the quality of the sequence. Below are the results from the Fastqc that was run on sample data:



**Figure 1:** **This report shows the Fastqc of the raw data.** Raw data shows that there are bases that are of poor quality, especially to the right side of the graph.

On the graph each box and whisker plot are created showing each position. The y axis shows the quality scores, the higher the score the better the base call is. The plots in the green indicates that it is good quality, orange represents okay quality and the red region represents poor quality (Andrews, 2010). *Fastq* files also calculate Phred scores, these are quality scores of the identification of the nucleotide bases of the sequence of DNA, it calculates the possible errors that could occur in regard to the sequence and identification of bases (Illumina, 2011: Wikipedia: Phred Scores, 2018). Phred scores are also used for *Trimmomatic* program, a minimum Phred score is created so that if there are Phred scores that are below the set minimum quality they will be trimmed off.

    After reviewing the quality given by the *Fastqc*, *Trimmomatic* is then used on the sequence to help fix any imperfection and problematic areas within the sequence. The *Trimmomatic* program is used to trim, filter, and clean up the sequence so it can be read and processed, and it also uses different algorithms to identify adapter sequences. There are two different techniques that Trimmomatic offers, but for the sample data the “Simple Mode” was used.

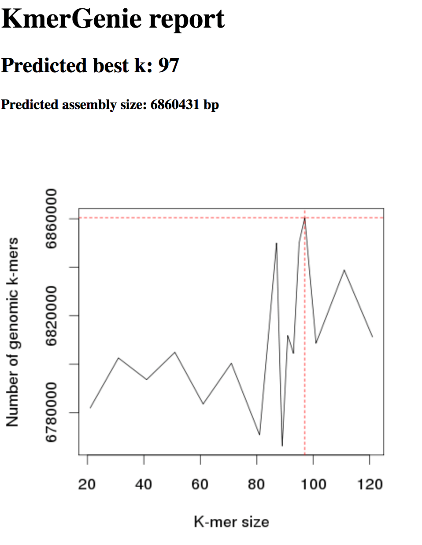


**Figure 2: This report shows the Fastqc of the data after it was ran through Trimmomatic.** In comparison to the raw data Fastqc this graph shows how the poor-quality regions were trimmed out and all of the base sequences are now good quality, making the overall sequence a high quality.

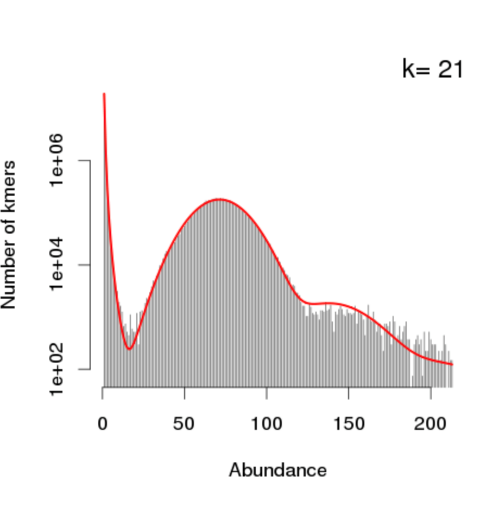
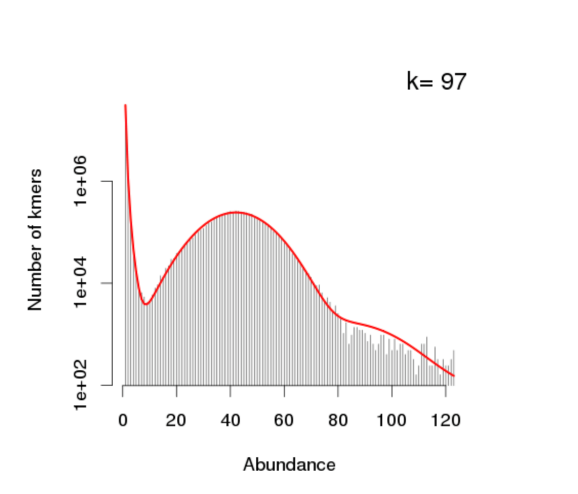
Simple Mode of Trimmomatic finds a similar match between the read and the sequence of the organism that is provided (Bolger, et al., 2014). The process starts by each of the reads are scanned from the 5’ to 3’ end to check if there were any adapters are there, then they are scanned to find any similarities between the technical sequences and the reads (Bolger, et al., 2014). It doesn't have to match completely, based on the match a local alignment is generated, if the alignment score is too high the aligned portion and the remainder past the aligned portion are removed (Bolger, et al., 2014: Li, et al., 2013).

**Genome Assembly- Assembly Size Estimation:**

    The data is then broken up into *K-mers*, K is the size of the fragments that gets cut up, there can be several different *K-mer* sizes, for this exercise we used *K-mers* that were 21 bases long and 97 bases long (Chikhi & Medvedev, 2014). *K-mers* are used to build *De Bruijn* graphs, *K-mers* are connected in the graph if they have the same or adjacent to another in a sequence (Melsted & Pritchard, 2011). This graph is showing the similarity within the *K-mers* and grouping them together. *K-mers* differ from one another by one base.  The *Kmergenie* program is used to predicts the assembly size of the genome. Below are the K-mer graphs that were generated using the *Kmergenie* program:



**Figure 3: KmerGenie Report**. Shows the number of *k-mers* and their size. The peaks indicate an increase in the number of *k-mers* relative to their size.



B

A

**Figure 4: K-mer graphs. A.)** The graph of *k-mers* that are 21 bases long. Indicating the amount of *k-mers* that size and their similarity to other *k-mers* that size. **B.)** The graph of *k-mers* that are 97 bases long. This was the *k-mer* size that was predicted to be the best and have the most similarity to one another that were this size.

The point of these graphs is to have the them be a normal bell-shaped curve. In the graph where the *k-mer* sizes are 21 the curve is more narrow, meaning there is more differences between the *k-mers*. While in the graph where the k-mers are 97 base pairs is more rounded showing there is less differentiation between the *k-mers*. The *k-mer* sizes that are 97 was calculated to be the best by *Kmergenie*, and it filtered out the errors.

**Genome Assembly- Assembly Algorithms:**

    The raw genomic sequence went from being broken up into fragments, to then being broken up into reads, then sized in *k-mers* and now *SOAPdenovo* will turn those *k-mers* into contigs and scaffolds. SOAPdenovo is used to build an assembly of all the data that has been filtered through quality control and cleaned up (Salzberg et. al., 2012). Contig is a group of overlapping DNA segments that form a region of DNA, they are used to create a map of the genome that is being sequenced (Del Angel, et. al., 2018). When two or more contigs are linked together incorrectly the overall assembly has larger contigs, which makes the assembly worse (Salzberg et. al., 2012). Scaffolds are made up of contigs and gaps that are chained together, they are used to help orient the sequence by position (Chin et. al., 2016). Scaffolds are often used for short read assemblies (Chin et. al., 2016). For *SOAPdenovo, k-mers* produce a N50 number per 100 base pairs and are used to fill in gaps, N50 is standard metric used to evaluate an assembly, it shows that 50% of the *k-mers* are bigger than the N50 value (Del Angel et. al., 2018). Below is the chart that was produced using different *k-mer* sizes:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Coverage** | **100X Trimmed** | | | **100X Untrimmed** | | |
| K-mer Size | N50 | Num\_Scaffs | Assembly Size | N50 | Num\_Scaffs | Assembly Size |
| 21 | 7240 | 25365 | 5,714,728 | 67705 | 181784 | 27,511,424 |
| 97 | 42702 | 1100 | 6,946,323 | 19553 | 2090 | 7,075,963 |

**Table 1. K-mer Comparison Chart.** The chart shows how increasing k-mer size decreases the number of scaffolds and increases the amount of nucleotides that contain at least 50% of the overall assembly length.

The table above shows that when the *k-mer* size increases the N50 increases as well. The overall assembly size also decreased when the *k-mer* size increased.

*MaSuRCA* (Maryland Super Read Cabog Assembler) is a program that is similar to *SOAPdenovo*, where it breaks the assembly into contigs to then get the N50 value and compare the assemblies that were made to see which one represents the genomic sequence the best (Zimin et. al., 2013). *MaSuRCA* also has a program built into it called *QuorUM*, this program is an error corrector, it corrects, trims, and removes reads that are not right or beneficial to the sequence. Below are the results from the *MaSuRCA:*

**N50= 147070**

|  |  |
| --- | --- |
| # of Contigs | MaSuRCA |
| > 0 base pairs | 576 |
| > 1,000 base pairs | 158 |
| > 5, 000 base pairs | 84 |
| > 10,000 base pairs | 74 |
| > 25,000 base pairs | 53 |
| > 50,000 base pairs | 41 |

**Table 2: MaSuRCA data.** 576 is the total number of contigs. As the contigs get bigger the MaSuRCA output gets smaller. The N50 is stating that 50% of the scaffolds are bigger than 147070.

*MaSuRCA* was able to take millions of reads, analyze them and then turn and group them into a total of 576 contigs. The N50 generated by MaSuRCA and the N50 generated by SOAPdenovo will be compared to one another, the N50 that is the highest shows the best quality of the assembly. The *MaSuRCA* showed the highest N50 value of 147070, which is much higher than the N50 of scaffolds that were produced in the *SOAPdenovo*, meaning that the *MaSuRCA* data has an assembly that is better quality.

**Genome Annotation- Repeat Finding:**

After the genome has been assembled, the process of gene annotation begins with identifying novel repeats within the genome. Genomes usually include a large portion of repeat sequencing called SINEs (short interspersed nuclear elements) and LINEs (long interspersed nuclear elements) which contain coding regions from exogenous sources such a virus. It is important that they are identified so the gene finders will not confuse SINEs and LINEs from endogenous genes that are necessary for the organism. SINEs and LINEs found in genomes, if not properly identified and masked, can lead to them being identified as rockfish genes or other genes totally unrelated to your organism’s genome. The *RepeatScout* program is the Linux tool to use to identify the repeats in the assembled genome. The program then filters out all the repeats and puts them into a labeled repeat library folder in Linux (Price et al. 2005). Once the repeats have been identified, the Linux tool, *RepeatMasker*, can be utilized to change repeats sequences into strands of N’s so the gene finder is unable to identify them as genes.



**Figure 5. Folder of Repeats Filtered from Sequence**. All of the novel repeat sequences are compiled in this folder after running *RepeatScout* on Linux.



**Figure 6. Folder of Genome Sequence**. This folder contains the whole genome sequence with the masked repeats. The repeats are masked with the letter N.

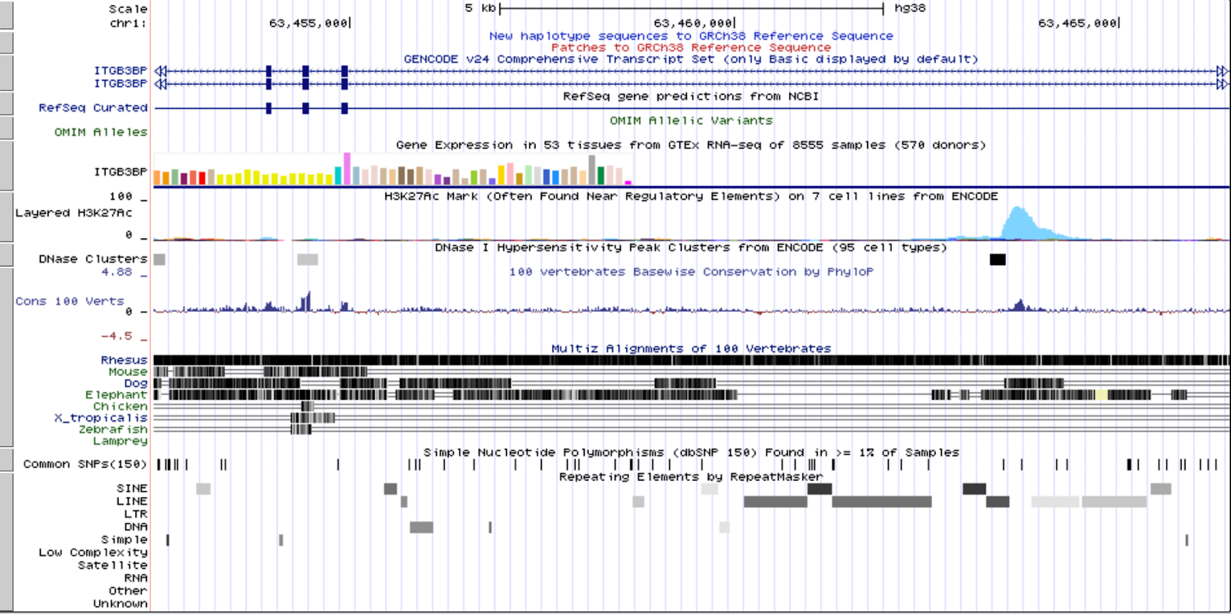
**Genome Annotation- *Maker*:**

Once the repeats in the genome sequence are masked, *Maker* can be run on Linux to generate a refined gene model more closely related and fit for the genome of interest. After *Maker* runs, it is important to train gene predictors given from the *Maker* results and have those gene predictors to be able to distinguish between different signals. *SNAP* and *Augustus* are gene predictor programs on Linux that have the ability to predict genes by locating intrinsic gene signals in DNA (Korf 2004). The start and stop codons and the intron and exon borders can be located with these programs (Korf 2004). *Fathom* is another program used after running *Maker* which creates a hidden-markov model (HMM) file that gives a summary about the information regarding the frequency of nucleotides at various areas within a single gene (Stanke and Waack 2003).

For further analysis of the genome in the FASTA file format, the Linux system offers a variety of command tools for genome annotation. The first one is called Grep. Grep can be used to grab certain rows of information stored in one or many files (Korf 2012). If rows are not needed for annotation, columns can be accessed from files using the *Awk* command (Korf 2012). Lastly, the BLAST (Basic Local Alignment Search Tool) command is able to do multiple types of alignments (nucleotide to nucleotide, protein to protein, protein to translated nucleotide alignment). BLAST can either be used to compare known databases online to the genome sequence of interest or it can be used with a custom database created by the BLAST program.

**SNP Calling and Interpretation:**

The next steps after genome annotation are identifying SNPs (single nucleotide polymorphisms) and visualizing the genome. A SNP is a variation in one nucleotide at a specific spot in a genome. It is also known as a valuable tool for finding genetic markers that reveal susceptibility to diseases (Fareed and Afzal 2013). The program *VARSCAN* in Linux is a tool in which SNPs can be identified. Then the SNPs can be compared to another organism of choice to see if they may contain the same SNPs. After comparing SNPs, the genome is ready to be visualized. There are many options available online to visualize a genome. Some examples include Integrated Genome Viewer and UCSC Genome Browser. They both are able to show the exons, introns, untranslated regions, SINEs, and LINEs of different genes within the genome.



**Figure 7. Gene Viewer.** An example of gene viewer tool available online is UCSC Genome Browser website.

**Conclusion:**

In conclusion, whole genome sequencing is the process of determining a DNA sequence of a known or unknown organism. It is an important research tool for scientists to get a better understanding of the organism of interest. However, a major pitfall of using Linux is that it is outdated. A command line system like Linux takes time to learn how to use properly and can be very frustrating for first time users. Commands need to be typed and spaced correctly. If not, programs will not run and/or errors will occur if a program runs with an improper command. In the future, it would great to see the creation of new operating system more user friendly for future biologists needing to sequence a whole genome.

**Works Cited:**

Andrews, Simon. 2010.FastQC: A quality control tool for high throughput sequence data. Reference Source.

Bolger, A. M., Lohse, M., and Usadel, B. 2014. Trimmomatic: A flexible trimmer for Illumina Sequence Data. *Bioinformatic*s, btu170.

Chin C.S., Peluso P., Sedlazeck F., et al. 2016. Phased Diploid Genome Assembly with Single Real-Time Sequencing. *Nature Methods*. doi: [10.1038/nmeth.4035](http://dx.doi.org/10.1038/nmeth.4035)

Chikhi R. & Medvedev P. 2013. Informed and automated k-mer size selection for genome assembly. Bioinformatics. NCBI. 30(1): 31-37

Del Angel V.D., Hjerde E., Sterck L., et al. 2018. Ten steps to get started in Genome Assembly and Annotation. F1000Research. ELIXIR. 7: 148

Fareed M, Afzal M. 2013. Single nucleotide polymorphism in genome-wide association of human population: A tool for broad spectrum service. Egyptian Journal of Medical Human Genetics. 14(2):123-134.

Illumina. 2011. Quality Score for Next-Generation Sequencing. Technical Note: Sequencing

Korf I. 2004. Gene finding in novel genomes. BMC Bioinformatics5:59

Korf I. 2012. Unix and Perl Primer for Biologists.

korflab.ucdavis.edu/Unix\_and\_Perl/unix\_and\_perl\_v3.0.pdf‎

Li J. W., Bolser D., Manske M., et al. 2013. The NGS Wikibook: a dynamic collaborative online training effort with long term sustainability. NCBI. 14(5): 548-555

Linux.com. 2016. What is Linux. *The Linux Foundation*. Available from: <https://www.linux.com/what-is-linux>

Price AL, Jones NC, Pevzner PA. 2005. De novo identification of repeat families in large genomes.Bioinformatics. 21:I351-I358 [repeat scout]

Salzberg SL, Phillippy AM, Zimin A, et al. 2012.  GAGE: a critical evaluation of genome assemblies and assembly algorithms. Genome Res 22: 557-567.

Stanke M, Waack S. Gene prediction with a hidden Markov model and a new intron submodel.Bioinformatics. 2003;19:ii215-ii225. doi: 10.1093/bioinformatics/btg1080.

Wikipedia. 2018. FastQ Format. Available from: <https://en.wikipedia.org/wiki/FASTQ_format>

Wikipedia. 2018. Phred Scores. Available from: <https://en.wikipedia.org/wiki/Phred_quality_score>

Zimin AV, Marçais G, Puiu D, Roberts M, Salzberg SL, Yorke JA. The MaSuRCA genome assembler. Bioinformatics. 2013. Nov 1;29(21):2669-77.