A. Introduction

My main goal was to do a full variant calling analysis of a breast cancer cell line, comparing variations found between a reference sequence and the tumor samples. By potentially finding novel or lesser known variations in genes, downstream analysis can then be used to identify annotations of those genes, their function, and add to the growing body of data of SNP's or other mutations in those genes that may be associated with breast cancer. I got the associated fastq files straight from here:

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM1172971

NOTE: There's a number of files I don't include in this folder because they are massive, up to 17GB.

B. Methods

Each step for the initial data extraction, QC, alignment, and variant calling is outlined in pipeline.sh, it's easiest to open it up side-by-side and compare the code to what I'm writing in this report.

NOTE: Don't run pipeline.sh unless you've got nothing to do for the next 24-48 hours, or are running it on a cluster. There are some extremely computationally intense steps in there.

a. Quality Control

To begin my analysis, my first step was to obtain some fastq files. The steps to do that are outlined in pipeline.sh under # getting data. This resulted in two 17GB fastq files, which I initially assumed were paired reads, but according to the link above, were single reads. To begin quality control, I went through several rounds of trimming, cutting out adapters, removing low quality reads, until finally I passed all my checks using fastqc. Each iteration can be found under /QC as an html file.

b. Alignment

Next, I began the important step of aligning, sorting, and indexing using a combination of bwa and samtools. I also marked duplicates using picard.jar.

c. Variant Calling

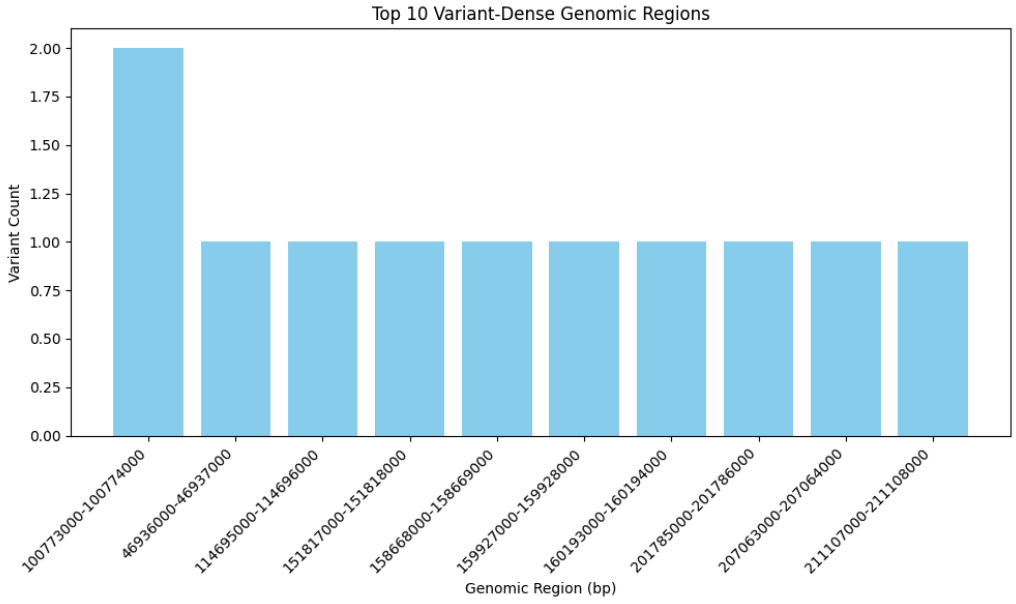
Then, I used GATK HaplotypeCaller to compare my downloaded reference to my sorted bam file using all of the indexes I created, both for the reference and the bam file itself.

d. Analysis

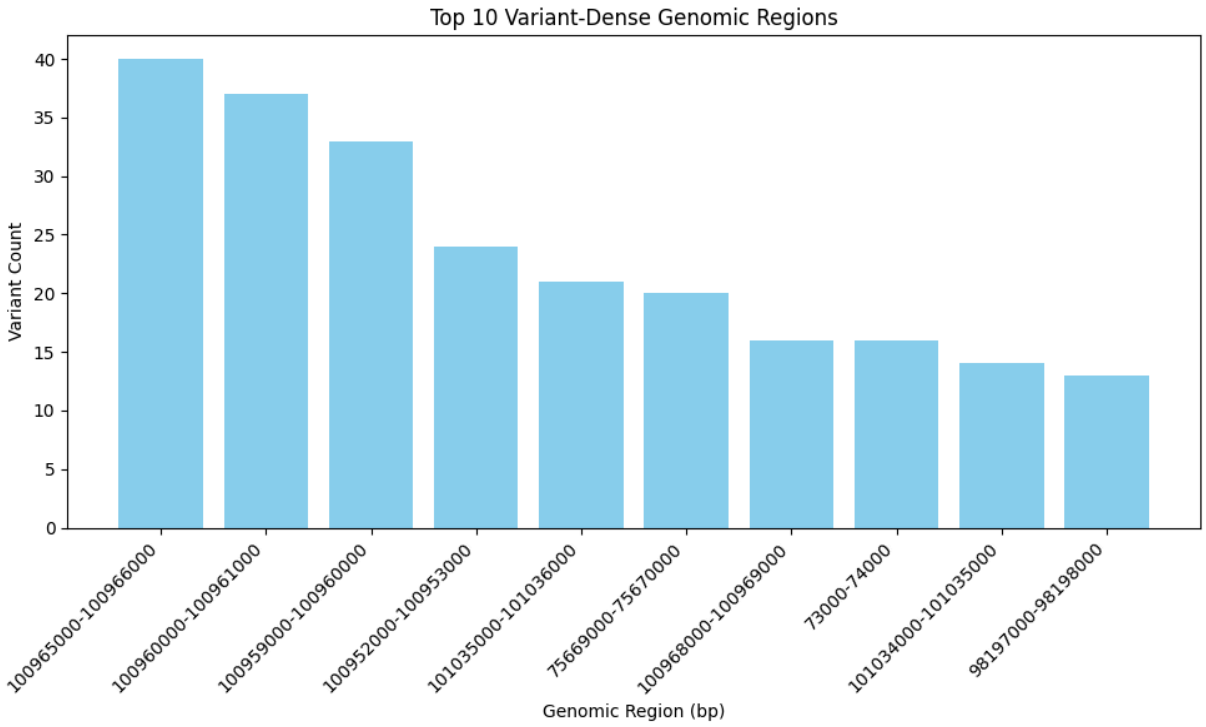
Finally, I wrote up a simple analysis pipeline in a file called variants.py, which is very easily runnable (unlike pipelines.sh). In this file, I used a .gtf file I found off of gencode to obtain annotations and cross-referenced that with my .vcf variants to see which genes were the most mutated. I also loaded up my .vcf variants into IGV to view some of the most mutated regions, which I was able to find using the first part of variants.py.

C. Results

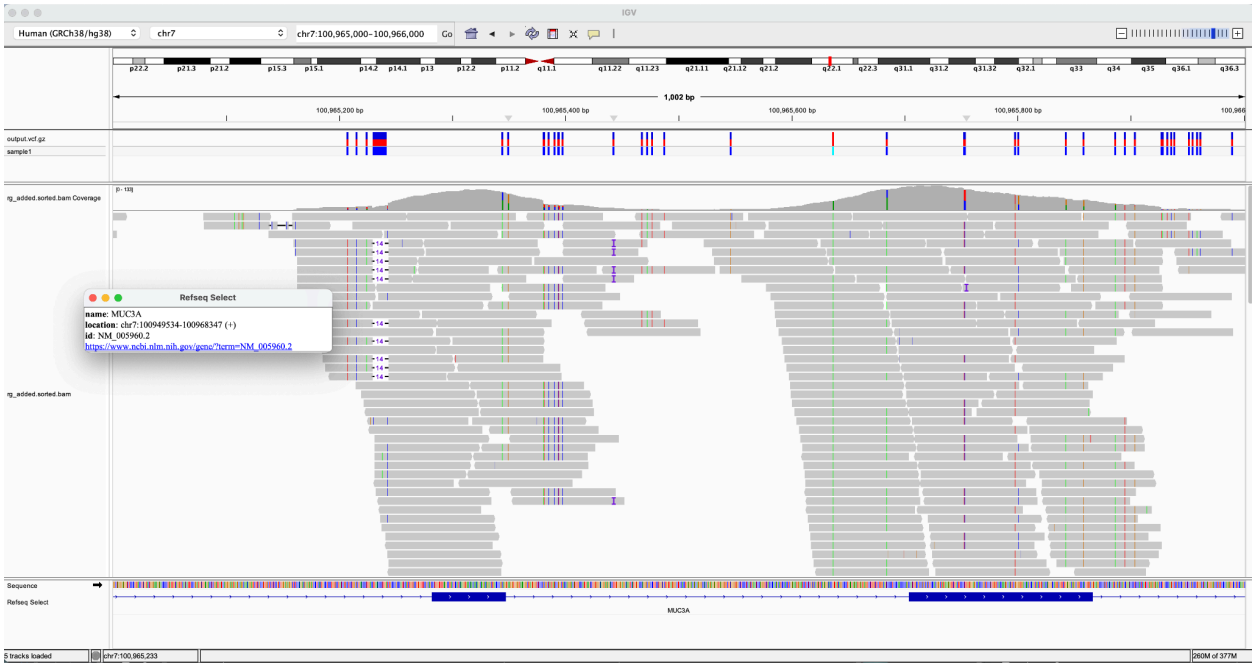
At first, I was interested in finding which regions and chromosomes had the most mutations. This depends on what the minimum amount of alternative alleles you consider. If you consider only more than one alternative allele, then most of the variants exist on chromosome 1, with two existing on chromosome 7, specifically on the ZAN gene, which functions as an adhesion protein between the sperm and egg, and is unlikely to play a role in breast cancer [1].



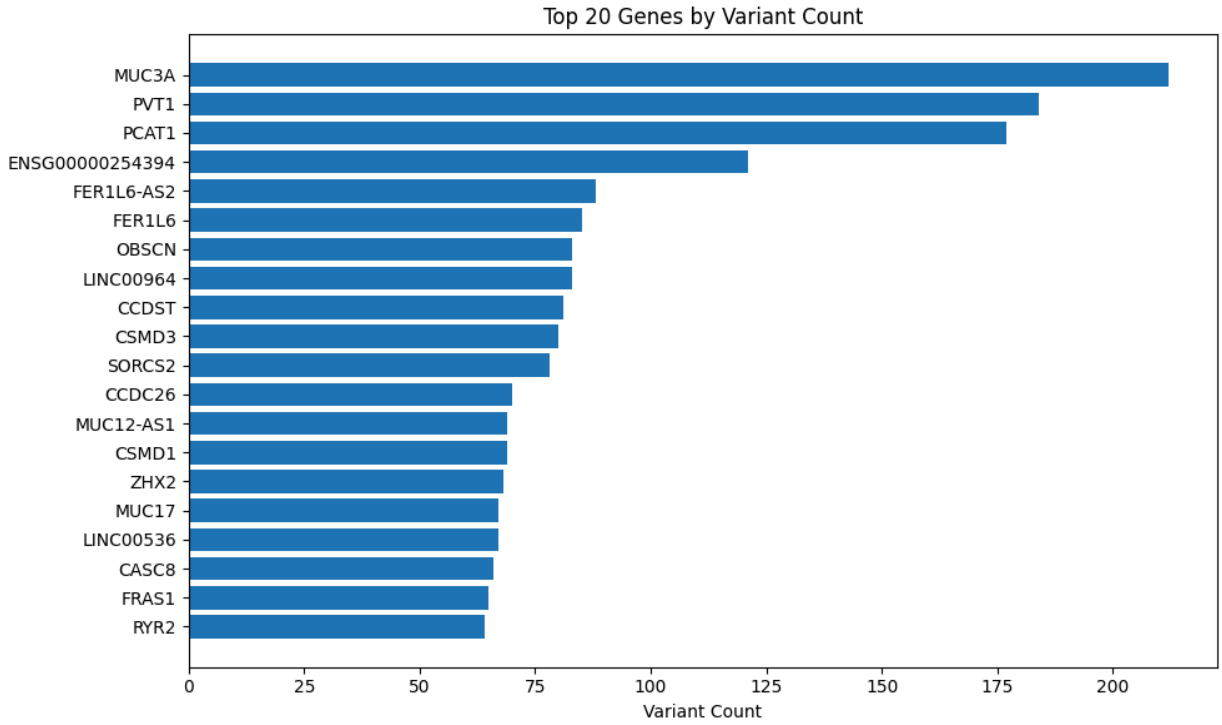
However, if you consider single alternative alleles, you get a much larger amount of variants



One of the most significant ones that I found was on a gene called MUC3A



MUC3A is an important gene involved in the production of mucus in epithelial cells, as well cell signaling, and other important processes [2]. It's mutation is also the cause of several other known cancers [2]. Based on these findings, I used the second part of the variants.py pipeline to find other genes that are heavily mutated



Not surprisingly, MUC3A was among the highest, followed by a few others with variant counts above 100. Among the variants was MUC17 and MUC12-AS1 which both play a similar role to MUC3A, underscoring the importance of the epithelial layer and cell-signaling to cancer development.

D. Discussion

Variant calling is an extremely important step in NGS analysis, and this analysis I did only really touches the surface. I looked at SNP's and indels, but beyond this, variant analysis can be used to better understand structural polymorphisms, gene fusions, gene homologies, and many more. There's also even more potential for a multi-layered analysis by looking at how protein and RNA data play into the mix, by observing how genetic changes make their way to alternative transcripts and actual expressed proteins. By using this analysis, researchers can find associations between current and developing medications, and how a particular patient's genetic profile may necessitate changes in their therapies. The hope is that this style of medicine will allow for a more individualized approach that more accurately targets the mechanism behind the disease.

E. References

[1] https://www.ncbi.nlm.nih.gov/gene/?term=NM_003386.3
[2] https://www.ncbi.nlm.nih.gov/gene/4584
[3] https://www.ncbi.nlm.nih.gov/gene/140453