2023-08-01

Merging Clinical and Genomic Data in Patients with Acute Leukemia for Downstream Analysis

Amanda Bataycan
University of Texas at El Paso

MERGING CLINICAL AND GENOMIC DATA IN PATIENTS WITH ACUTE LEUKEMIA FOR DOWNSTREAM ANALYSIS

AMANDA M. BATAYCAN

Master's Program in Computational Science

APPROVED:

Ming-Ying Leung, Ph.D., Chair

Jonathon Mohl, Ph.D.

Georgialina Rodriguez, Ph.D.

Stephen L. Crites, Jr., Ph.D.
Dean of the Graduate School

MERGING CLINICAL AND GENOMIC DATA IN PATIENTS WITH ACUTE LEUKEMIA FOR DOWNSTREAM ANALYSIS

by

AMANDA MARIA BATAYCAN,

MS in Bioinformatics

BS in Biology and Applied Mathematics

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

AUGUST 2023

Acknowledgements

Since I began my journey at UTEP I have had the greatest support system, both from faculty and family, for that I would like to express my absolute gratitude.

To my research advisor and mentor, Dr. Leung, for her guidance, motivation, patience, and priceless knowledge she has shared with me throughout the years. I can never thank her enough for all that she has helped me with from my first day to now. To Dr. Mohl, from the department of Mathematical Sciences, an excellent professor who genuinely wants his students to reach their full potential. His experience and expertise in computing is a treasure and I am very grateful to have had his help and support throughout this project.

To my parents, for their endless love and support, they have taught me to face all challenges with courage and an open mind. Finally, my niece, Jay, who has been my light through every dark day and reminded me that no matter how hard today is, tomorrow will be better.

Abstract

The purpose of this study is to integrate multiple sources of information from patients with acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) to construct organized datasets that would enable downstream bioinformatics and statistical analyses of the patients' survival status and overall survival times in relation to their demographic, clinical, and genomic mutation profiles.

With NIH Genomic Data Commons as the primary data resource and cBioPortal as the access portal, datasets on 149 and 603 unique patients with AML and ALL, respectively, were obtained. Python scripts were written to compile individual patients' single nucleotide variant (SNV) data files into one dataset for each patient group. In both groups, over 95% of the SNVs occurred only in tumor samples while less than 0.02% only in normal samples. Compared to normal variants, tumor SNV change types favored mutations that reduced GC content of genes in both patient groups. Additional results showed shifts of variant densities on all chromosomes, most noticeably on chromosome 11 in patients with AML and chromosome 2 in patients with ALL.

One important task accomplished in this work was merging the individual patients' SNV data with their corresponding demographic and clinical information, which includes ethnicity and race, disease classification or staging, as well as survival outcomes among other variables. With the merged data, we propose several bioinformatics studies to investigate the functional effects of SNVs and to select likely leukemia-associated genes not reported to date in published literature. SNV occurrence frequencies in the selected genes will augment the patients' demographic and clinical information to form the final set of variables to be analyzed. Our goal is to establish a predictive model for patients' overall survival times to facilitate discoveries of potential gene therapy targets for acute leukemia.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

With the technological developments just in the last two decades, communities across the world have been able to work together to push further than previous generation could have imagined. Using next generation sequencing on collected samples, biotech laboratories can determine the nucleotide sequences in DNA molecules, which are then written to computer-readable files. Once generated, these data files can be published for others to utilize for various purposes. With the vast amount of data currently available, one can use computational tools to extract useful information and organize them for downstream analyses to help answer many scientific and biomedical questions, but the process is usually very tedious and time consuming. Fortunately, with the advancements in computing technology these tasks have become more efficient in time and quality. Focusing on abnormalities found in the available DNA sequences from patients with cancer, our goal is to help identify a causation and ultimately a resolution for this disease. In this thesis, I will describe my work in organizing integrated datasets for two types of acute leukemia in preparation for downstream bioinformatics analysis.

1.1 MUTATION IN DNA

There are two ways a mutation in DNA can arise, naturally or through an external component. The natural occurrence of a mutation is seen during cellular division, the DNA is copied and in some cases the copy taken is not an exact replica. As for the external components, exposure to radiation and other chemicals can increase the levels of reactive oxygen species (ROS) which are harmful to the body and all major components of the cell. ROS are formed when a molecule containing oxygen gains or losses an electron, causing it to become negatively charged and ultimately interfering with a number of different pathways, including DNA replication [13, 14].

One type of mutation that can conduce to cancer is termed single nucleotide variant (SNV), where an alteration of a single base in the nucleotide sequence occurs. If an SNV occurs on a protein-coding DNA segment, there will be either a synonymous or nonsynonymous change in the amino acid sequence during translation. Synonymous changes arise when the nucleotide bases changes, but the amino acid sequence remains the same, whereas nonsynonymous changes lead to changes in both the nucleotide sequence and amino acid sequence. By altering the amino acid sequence in cancer-related genes (e.g., oncogenes that may transform a normal cell to tumor cell, or tumor suppressant genes that are responsible for protecting the body from cancer), a functional effect that results in the reduction or total loss of the protein's normal function may be observed.

Leukemia, or cancer of blood cells, can be a rapid or stagnant growing disease. The growth primarily depends on the affected blood cells and the symptoms are more obscure than those in solid tumors. From previous studies throughout the year, leukemia, has shown to be more prominent in patients over the age of fifty-five or children under the age of 15. With an incredible amount of research spent on this topic, the National Cancer Institute estimated the five-year survival rate at 65.7% [2]. Depending on the type of leukemia, patients can be classified using staging system that physicians define based on the current progression of the cancer. This can be useful when determining a patient's prognosis and treatment [15].

Leukemia can be broken into two groups: Myeloid and Lymphoblastic. Both of which can have the prefix as acute or chronic. Chronic Leukemia is one in which the mutation arises in a mature blood cell, that has had time to function normal prior to the mutation, versus, Acute Leukemia where the mutation is in a young blood cell. Due to the shortened normal functioning period of the altered young blood cell, the acute form of leukemia is much more aggressive.

1.2 ACUTE LEUKEMIA

As one of the most common leukemias in the adult population, acute myeloid leukemia (AML) originates in early forms of myeloblast, red blood cells, and platelets. Responsible for approximately 4% cancer related deaths in 2022, this form of cancer has mixed prognosis with a 66% five-year survival rate [2]. Although AML research has advanced, unfortunately those diagnosed over the age of 60 have a lower survival rate [3].

Unlike the traditional classification system used in solid tumors, due to the diversity of myeloid cells that the mutation occurs in, AML uses the French, American, and British (FAB) classification to categorize these outcomes. Ranging from M0 through M7, this system was developed in the 1970s using a cell staining methodology and gave an idea the type and maturity of the cell being altered [1]. Similarly, to solid tumor cancers, AML can also be classified as primary or secondary, using it to identify the order of metastasized tumors. Due to the multidrug resistance mechanisms affected, as a result, most cases of secondary AML has shown to have a poor prognosis [4]. An example of primary AML is one that originates in the bone marrow allowing it to spread to other vital organs, including the liver and spleen [1], this also reduces the likelihood of a positive prognosis by causing other organ failures.

The table below displays the different FAB subtypes that occur in AML and their corresponding morphological classifications. The classifications are named based on the damaged cell type within the myeloid stem cell lineage. This can be determined using a blood stain test.

Table 1.1: FAB subtypes and their morphological classifications.

FAB Subtype	Morphological Classification
M0	Undifferentiated acute myeloblastic leukemia
M1	Acute myeloblastic leukemia with minimal maturation
M2	Acute myeloblastic leukemia with maturation
M3	Acute promyelocytic leukemia
M4	Acute myelomonocytic leukemia
M5	Acute monocytic leukemia
M6	Acute erythroid leukemia
M7	Acute megakaryoblastic leukemia

In contrast to AML, acute lymphoblastic leukemia (ALL) is most common in children. ALL originates in early forms of lymphocytes, which can be broken into two different types: B-cells and T-cells, each having separate functions in the body's immune defenses. B-cells are responsible for attacking foreign bodies that have entered blood stream by latching on, whereas T-cells are responsible for destroying the body's own cells that have become cancerous or infected from another foreign body [5].

One of the repercussions of ALL is the spreading of the diseases in the central nervous system (CNS). While this is not typically observed in the initial diagnosis of patients with ALL, its presence is one of the most severe complications that can occur and has been found to significantly reduce the overall survival. The CNS involvement is most detected in ALL patients who have relapsed, at approximately 30%, this event usually results in treatment failures [26].

By collecting a sample of cerebral spinal fluid, the CNS involvement can be categorized by the number of white blood cells and leukemia blasts that are present [27]. The table below displays the classification of the CNS involvement by the three main subtypes [28].

Table 1.2: CNS stage types with their ranges of WBCs and presences of Leukemia Blasts.

CNS Stage	**White Blood Cells (WBC) Description**	**Leukemia Blasts**
CNS 1	<= 5/mL	Absent
CNS 2	<= 5/mL	Present
CNS 3	> 5/mL	Present

1.3 RESEARCH OBJECTIVE

The ultimate goal of this research is to identify and understand common trends in the SNV profiles in a variety of patients with leukemia and eventually generate an integrated predictive model that can help to design personalized treatment regimens of individual patients. Within the scope of this thesis, my specific aims are:

(1) Review published literature and databases to look for data collected from patients with AML and ALL, and compile a list of genes currently known to be associated with them.

(2) Organize integrated datasets for patients with AML and ALL that include their demographic, clinical, and SNV data.

(3) Conduct statistical overviews for the organized patient data to facilitate downstream bioinformatics analysis to be conducted as your PhD dissertation research.

Chapter 2: Literature Review

2.1 SNV AND GENES ASSOCIATED WITH ALL AND AML

For years the traditional chemotherapy using cytotoxic agents has been the primary treatment for AML and ALL. However, with the vast information and advancements in sequencing the human genome, researchers have been able to utilize this information to develop targeted gene therapies for both. Target gene therapy is less invasive on the body than chemotherapy and has been shown to be more responsive [8]. With the goal of either replacing, or enhancing traditional chemotherapy, next generation sequencing (NGS) has facilitated great progress in target gene therapies for these patients. NGS technology determines genetic variation, one of those being SNVs, and their association to diseases. Advantages to NGS is its ability to detect the variations with a lower input requirement while still maintaining high accuracy by being able to identify these variants even at low allele frequencies, making the performance quick and cost effective [29].

The crucial step in developing the targeted gene therapies is the use of NGS data to find frequently occurring genetic mutation commonalities, including SNVs, among patients and ultimately understanding their specific pathways. An example of a registry that has been developed utilizing this data is the American Association for Cancer Research (AACR) Genomics, Evidence, Neoplasia, Information, Exchange (GENIE) project. Affiliated with over 20 participating cancer institutions, within the states and internationally, AACR project GENIE contains 167,000 sequenced samples from over 148,000 patients, with leukemia being one of the top eleven frequent cancers [30]. The data files of these patients can be directly downloaded from Sage Bionetworks, which is an organization that promotes scientifical practices and patient engagement into the researching process. The data was also imported into cBioPortal, a portal that will be discussed in detail later being that it is a primary resource that was used within this study, which it can also be accessed within this site. However, to access the data information on the following AML and ALL that was used in AACR project GENIE, a user must be granted authorization, so the background information on their findings are limited. Using the AACR project GENIE, it was found that the most prevalent gene alterations for ALL were WT1, NOTCH1, EZH2, BCORL1, and USP7. These alterations include a variety of mutation types from missense (nucleotide base change in sequence) to frame shifts (insertion or deletion of a nucleotide base in sequence) [7], both of which are subtypes of an SNV that alter the amino acid sequence during translation making them nonsynonymous mutations.

AACR project GENIE found the corresponding data for AML. They determined that AML had high mutation rates in DNMT3A, FLT3, TP53, NPM1, and RUNX1 [9]. Other studies have also confirmed that mutations in FLT3 have shown to have a correlation to AML [8]. In both studies, they review already FDA approved targeted gene therapies to have shown to be highly effective for these patients.

Previously mentioned about the external environmental risks being a factor in the causation of mutations, one article investigated the effects of a thyroid cancer treatment plan has on developing AML. This article reports that after receiving a single dose of Iodine-131 isoform, patients experienced a single base substitution resulting in a RAS gene mutation [10]. Aftereffects of this mutation leads to uncontrolled proliferation and blocking the cell from apoptosis.

2.1.1 AML Associated Genes

Compiled from several published articles within the last 10 years, a list of genes that have been found to have alteration in patients with AML. This list contains 93 different genes, many of these articles highlighted the importance of the FLT3, NPM1, CEBPA, ASXL1, RAS and IDH groups as key contributors. Meaning that the mutations that arose in these genes created a cascading effect that resulted in the malfunction of the other listed genes. Nonetheless, to validate the research found in this study all the genes that had any found link to AML were annotated.

2.1.2 ALL Associated Genes

A similar literature approach to the list generated for the AML-associated genes was done for the associated genes with patients with ALL. The ALL-associated gene list obtained from other published sources contained 89 unique genes. Again, these articles emphasize that the key contributing genes as the RAS group, IKZF1, PAX5, EZH2, and MEF2D which was also found in the AML associated genes list, create the expression level changes in later genes.

2.2 DATA PORTALS AND DATA REPOSITORIES

With the rapid advancements in technology over the past few decades, vast resources for research in multiple fields have also grown exponentially. In the biomedical areas, we have seen enormous developments ranging from large public databases sharing intertwined information, to software and their limitless packages that are being added and updated continuously. Portals, such as the National Cancer Institute Genomic Data Commons (GDC) and cBioPortal for Cancer Genomics, are developed to facilitate in the storage of this information to be easily found and downloaded.

An example of this type of relationship previous mentioned was AACR GENIE, where the data is now readily available on cBioPortal. Additional examples of this are also The Cancer Genome Atlas Program (TCGA) and Therapeutically Applicable Research to Generate Effective Treatments (TARGET) [32, 33]. Both of which, have their clinical or mutational data on GDC and cBioPortal, and share the common goal of utilizing and distributing information to better treatments and improve patient prognosis.

2.2.1 Data Portals

By supporting several cancer genome projects, GDC, provides the capability to upload information, thus aiding in the exploration of this shared data to cancer researching communities. To oblige this mission, GDC contains the ability to upload, access, and analyze data into their portal. In total, they provide the researchers and institutions with over 100 programs and projects and have over 86,000 cases that can be accessed [34].

Particularly useful for my research is the large collection of SNV records of individual patients with different types of cancer obtained from various NGS projects. When a specimen is collected from normal and tumor tissue sample of a patient, these samples can be sequenced and the mutations deviating from the reference human genome are annotated and stored as variant call format (VCF) files that can be accessed and downloaded by registered members of GDC. However, these VCF files contain only mutational data but not demographic information. The only available information about the patient is a patient ID assigned by the group conducting the sequencing.

Originally developed at Memorial Sloan Kettering Cancer Center, a cancer treatment and research institution located in New York, cBioPortal is now managed and maintained by multiple cancer-related institutions within the United States and internationally. With a similar focus as GDC, by bridging the genomic data to external cancer researchers, they also provide molecular profiles and clinical attributes from the genomic projects that are being submitted. Additional tools on this site, allow users to build their own visualizations and reports by connecting to their application programming interface (API) [35].

The cBioPortal for Cancer Genomics (cBioPortal), contains mutational data which requires authorization to obtain certain datasets, but conveniently has demographic detail about the patients that are downloadable for all users. To protect the patient’s privacy, no personal information is provided, just demographic factors that could contribute to types of cancers is known. These factors can include age, gender, ethnicity, stage of cancer, and some overall survival rates. Additional information given such as treatment and remission or relapse can be potentially known, however some demographic details may not be available for all cancer types.

2.2.2 Data Repositories

Data repositories, also referred to as programs or projects, are organized projects that conduct their own research independently with their own specific focuses. Once finalized they upload their analysis onto Data Portals, along with the data they have collected to form this analysis. Typically, the well-established projects focus is on genomic results for therapeutic enhancements on current forms of treatments.

An example of this would be the previously mentioned TARGET and TCGA projects. TCGA, is one that was found to have three different sub-studies performed using the data that was uploaded onto the cBioPortal. The three sub-studies in this case were the New England Journal of Medicine (NEJM), PanCancer Atlas, and Fire Hose, all of which uploaded their own findings onto cBioPortal. By allowing the public to obtain these findings, it creates an opportunity for fellow researchers to build upon similar topics and expand into a new perspective.

2.3 Programming tools

When handling large data, format for any software or program is crucial due to the specificity of the input information. As any researcher dealing with big data analysis has found, having properly organized data is of paramount importance. After exploring different possibilities, I have found that a combination of Python, Microsoft Excel, and R would be optimal for organizing the data from multiple sources into a format that will be best suited for our planned downstream bioinformatics analysis. Python is preferred for its capabilities of manipulating text strings, which is essential for extracting information from the individual VCF files downloaded from the public databases. Excel is the most convenient tool for manual examination of the data files, both before and/or after processing. R provides the most comprehensive functions and packages for statistical analysis and visualization.

The features that were used in organized the data using python was a package called Pandas, that included being able to read in a csv file and convert it to a dataframe, or a virtual table formatted data type in software programming, merging multiple dataframes together based on column(s) common entries, remove duplicated information, and finally being able to extract any modified dataframe back into a new csv file. When using R for the statistical features, it is an easier application to use for finding unique values, creating, and exporting tables, and generating box plots and other types of graphs for interpretation. Excel allows for an easier readability of the csv file and a faster resource for finding certain values and generating linear and bar graphs.

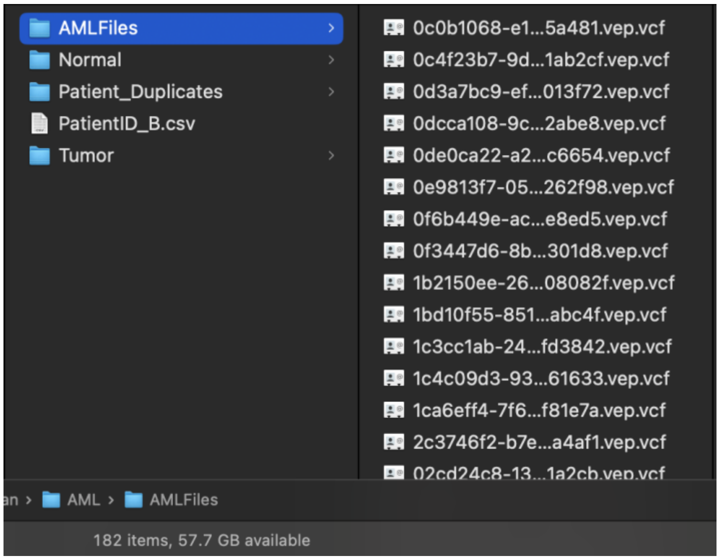
Chapter 3: Materials and Methods

Mentioned previously the different portals in which databases were extracted and how they were constructed all were based on certain variant criteria. Since there were two separate extractions from different portals the information that needed to be condensed into a single file. Once this was completed, there was a large dataset with many variables and information so an exploratory data analysis could be conducted.

3.1 DATA EXTRACTION AND CONSTRUCTION

From the GDC portal, variant call format (VCF) files were downloaded on both AML and ALL search criteria. Different filtration parameters can be introduced in this step including, the mutation type (i.e. SNV or motif), file type, ethnicity, etc. Although this web-based portal contains various types of mutations, our specific search was focused on only the single nucleotide variants (SNVs) and that they were VCF files. With this, 182 VCF files for AML and 620 VCF files for ALL, were downloaded. Using PyCharm these VCF files were read into a outsourced Python script [23], that converted the VCF files into comma-separated values (CSV) files and implemented a criteria that only extracted variants that would be most informative, which was concluded based on their allele depth (AD). More detail on AD is provided below.

The purpose behind reorganizing the data into CSV files was to create easier readability for the user, as well as compatibility when integrating it into programming scripts. Below are figures for the main AML and ALL folders. Once downloaded the VCF files were stored into a subfolder on a local computer for the conversion script to operate on. For each VCF file, the script outputs two separate CSV files, one normal and one tumor. These CSV files were then stored into their corresponding subfolders “Normal” and “Tumor”.

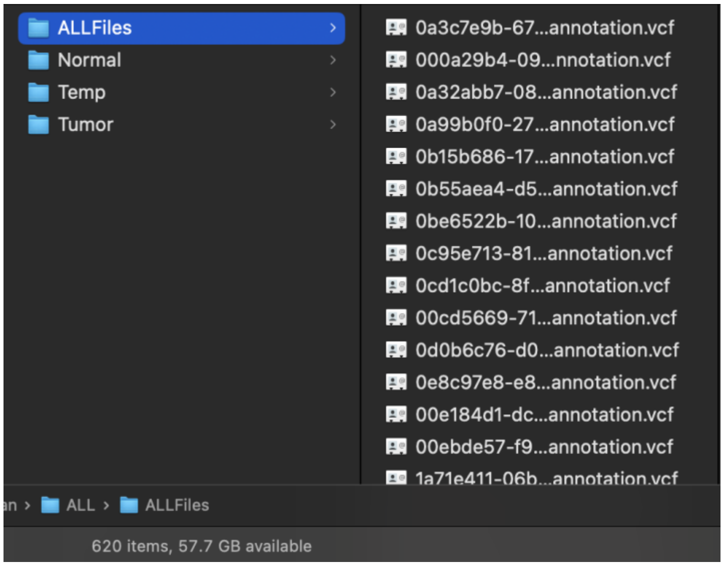


```
AMLFiles
Normal
Patient_Duplicates
PatientID_B.csv
Tumor
```

```
0c0b1068-e1...5a481.vep.vcf
0c4f23b7-9d...1ab2cf.vep.vcf
0d3a7bc9-ef...013f72.vep.vcf
0dcca108-9c...2abe8.vep.vcf
0de0ca22-a2...c6654.vep.vcf
0e9813f7-05...262f98.vep.vcf
0f6b449e-ac...e8ed5.vep.vcf
0f3447d6-8b...301d8.vep.vcf
1b2150ee-26...08082f.vep.vcf
1bd10f55-851...abc4f.vep.vcf
1c3cc1ab-24...fd3842.vep.vcf
1c4c09d3-93...61633.vep.vcf
1ca6eff4-7f6...f81e7a.vep.vcf
2c3746f2-b7e...a4af1.vep.vcf
02cd24c8-13...1a2cb.vep.vcf
```

```
an > AML > AMLFiles
182 items, 57.7 GB available
```

Figure 3.1: AML main folder, containing subfolders: “AMLFiles” with all 182 extracted AML VCF files, “Normal” with 182 normal CSV files, and “Tumor” with 182 tumor CSV files.



```
ALLFiles
Normal
Temp
Tumor
```

```
0a3c7e9b-67...annotation.vcf
000a29b4-09...nnotation.vcf
0a32abb7-08...annotation.vcf
0a99b0f0-27...annotation.vcf
0b15b686-17...annotation.vcf
0b55aea4-d5...annotation.vcf
0be6522b-10...annotation.vcf
0c95e713-81...annotation.vcf
0cd1c0bc-8f...annotation.vcf
00cd5669-71...annotation.vcf
0d0b6c76-d0...annotation.vcf
0e8c97e8-e8...annotation.vcf
00e184d1-dc...annotation.vcf
00ebde57-f9...annotation.vcf
1a71e411-06b...annotation.vcf
```

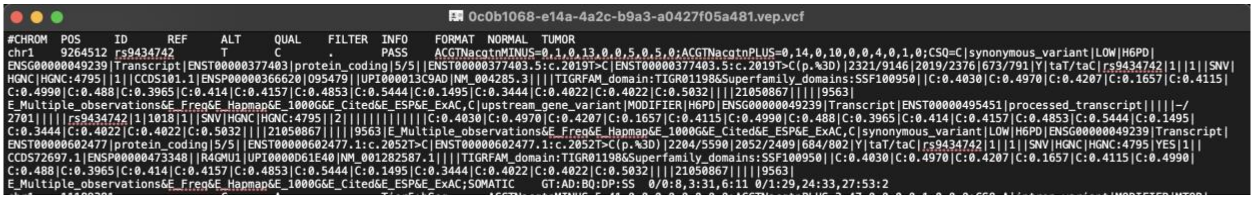
```
an > ALL > ALLFiles
620 items, 57.7 GB available
```

Figure 3.2: ALL main folder, containing subfolders: “ALLFiles” with all 620 extracted ALL VCF files, “Normal” with 620 normal CSV files, and “Tumor” with 620 tumor CSV files.

In transforming the VCF files to CSV format, the main parameter that needs to be set was the allele depth (AD) for screening the SNVs in the VCF files. AD is defined as the number of reads that support each reported allele and a low AD would identify the SNV to be uninformative.

Using this as a filtering parameter will retain the variants that have enough statistical evidence to continue for downstream analysis. This information is stored within the VCF along with chromosomal, positions, change types, and additional annotations, but these other values will not be used as variant screening criteria.

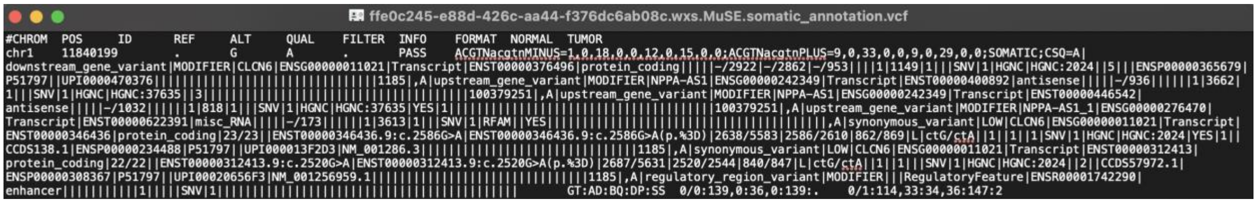
Figures 3.3 and 3.4 display the top lines of respective VCF file examples for AML and ALL, opened with a text editor. Each figure contains the header in the first row, and the first variant in the example file in the subsequent rows. The entries for different columns are tab-separated. The previously described AD values are initially found and read within this stage by using the FORMAT, NORMAL, and TUMOR columns. FORMAT column identifies how this entry is to be read, AML and ALL having the same format with GT:AD:BQ:DP:SS, meaning that each value within the entries for the NORMAL and TUMOR columns will be separated by colons. Focusing on the AD value only, the script reads the values after the first colon within the NORMAL and TUMOR columns.



0c0b1068-e14a-4a2c-b9a3-a0427f05a481.vep.vcf

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  NORMAL  TUMOR
chr1    9264512 rs9434742       T       C       .       PASS    ACGTNacgtnMINUS=0,1,0,13,0,0,5,0,5,0;ACGTNacgtnPLUS=0,14,0,10,0,0,4,0,1,0;CSQ=C|synonymous_variant|LOW|H6PD|
ENSG00000049239|Transcript|ENST00000377403|protein_coding|5/5||ENST00000377403.5:c.2019T>C|ENST00000377403.5:c.2019T>C(p.%3D)|2321/9146|2019/2376|673/791|Y|taT/taC|rs9434742|1||1||SNV|
HGNC|HGNC:4795||1||CCDS101.1|ENSP00000366620|O95479||UPI000013C9AD|NM_004285.3||||TIGRFAM_domain:TIGR01198&Superfamily_domains:SSF100950||C:0.4030|C:0.4970|C:0.4207|C:0.1657|C:0.4115|
C:0.4990|C:0.488|C:0.3965|C:0.414|C:0.4157|C:0.4853|C:0.5444|C:0.1495|C:0.3444|C:0.4022|C:0.4022|C:0.5032||||21050867|||||9563|
E_Multiple_observations&E_Freq&E_Hapmap&E_1000G&E_Cited&E_ESP&E_ExAC,C|upstream_gene_variant|MODIFIER|H6PD|ENSG00000049239|Transcript|ENST00000495451|processed_transcript|||||-/
2701|||||rs9434742|1|1018|1||SNV|HGNC|HGNC:4795||2|||||||||||C:0.4030|C:0.4970|C:0.4207|C:0.1657|C:0.4115|C:0.4990|C:0.488|C:0.3965|C:0.414|C:0.4157|C:0.4853|C:0.5444|C:0.1495|
C:0.3444|C:0.4022|C:0.4022|C:0.5032||||21050867|||||9563|E_Multiple_observations&E_Freq&E_Hapmap&E_1000G&E_Cited&E_ESP&E_ExAC,C|synonymous_variant|LOW|H6PD|ENSG00000049239|Transcript|
ENST00000602477|protein_coding|5/5||ENST00000602477.1:c.2052T>C|ENST00000602477.1:c.2052T>C(p.%3D)|2204/5590|2052/2409|684/802|Y|taT/taC|rs9434742|1||1||SNV|HGNC|HGNC:4795|YES|1||
CCDS72697.1|ENSP00000473348||R4GMU1|UPI0000D61E40|NM_001282587.1||||TIGRFAM_domain:TIGR01198&Superfamily_domains:SSF100950||C:0.4030|C:0.4970|C:0.4207|C:0.1657|C:0.4115|C:0.4990|
C:0.488|C:0.3965|C:0.414|C:0.4157|C:0.4853|C:0.5444|C:0.1495|C:0.3444|C:0.4022|C:0.4022|C:0.5032||||21050867|||||9563|
E_Multiple_observations&E_Freq&E_Hapmap&E_1000G&E_Cited&E_ESP&E_ExAC;SOMATIC    GT:AD:BQ:DP:SS  0/0:8,3:31,6:11 0/1:29,24:33,27:53:2
```

Figure 3.3: Example of variant information found on an AML VCF file.

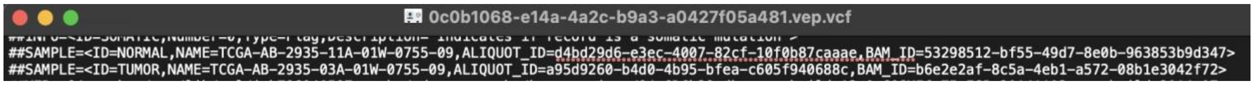


ffe0c245-e88d-426c-aa44-f376dc6ab08c.wxs.MuSE.somatic_annotation.vcf

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  NORMAL  TUMOR
chr1    11840199        .       G       A       .       PASS    ACGTNacgtnMINUS=1,0,18,0,0,12,0,15,0,0;ACGTNacgtnPLUS=9,0,33,0,0,9,0,29,0,0;SOMATIC;CSQ=A|
downstream_gene_variant|MODIFIER|CLCN6|ENSG00000011021|Transcript|ENST00000376496|protein_coding||||-/2922|-/2862|-/953||||1|1149|1|||SNV|1|HGNC|HGNC:2024||5|||ENSP00000365679|
P51797||UPI0000470376|||||||||||||||||||||||||||1185|,A|upstream_gene_variant|MODIFIER|NPPA-AS1|ENSG00000242349|Transcript|ENST00000400892|antisense||||-/936|||||1|3662|
1|||SNV|1|HGNC|HGNC:37635||3||||||||||||||||||||||||||||||||||100379251|,A|upstream_gene_variant|MODIFIER|NPPA-AS1|ENSG00000242349|Transcript|ENST00000446542|
antisense||||-/1032||||||1|818|1|||SNV|1|HGNC|HGNC:37635|YES|1||||||||||||||||||||||||||||||||||100379251|,A|upstream_gene_variant|MODIFIER|NPPA-AS1_1|ENSG00000276470|
Transcript|ENST00000622391|misc_RNA||||-/173||||||1|3613|1|||SNV|1|RFAM||YES|||||||||||||||||||||||||||||||||||||,A|synonymous_variant|LOW|CLCN6|ENSG00000011021|Transcript|
ENST00000346436|protein_coding|23/23||ENST00000346436.9:c.2586G>A|ENST00000346436.9:c.2586G>A(p.%3D)|2638/5583|2586/2610|862/869|L|ctG/ctA||1||1||1|SNV|1|HGNC|HGNC:2024|YES|1||
CCDS138.1|ENSP00000234488|P51797||UPI000013F2D3|NM_001286.3||||||||||||||||||||||||||1185|,A|synonymous_variant|LOW|CLCN6|ENSG00000011021|Transcript|ENST00000312413|
protein_coding|22/22||ENST00000312413.9:c.2520G>A|ENST00000312413.9:c.2520G>A(p.%3D)|2687/5631|2520/2544|840/847|L|ctG/ctA||1||1|||SNV|1|HGNC|HGNC:2024||2||CCDS57972.1|
ENSP00000308367|P51797||UPI00020656F3|NM_001256959.1||||||||||||||||||||||||||||1185|,A|regulatory_region_variant|MODIFIER|||RegulatoryFeature|ENSR00001742290|
enhancer||||||||||1|||||SNV|1|||||||||||||||||||||||||||||||||||||||||        GT:AD:BQ:DP:SS  0/0:139,0:36,0:139:.    0/1:114,33:34,36:147:2
```

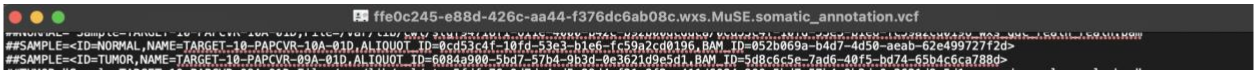
Figure 3.4: Example of variant information found on an ALL VCF file.

For both AML and ALL, each VCF file has a unique sample ID taken from a patient, but VCF files with different sample IDs could come from the same patients. It is assumed that during extraction of these samples, partitions were made to do multiple analysis on each main sample and the sample ID that was given is the identity of each part. However, the patient ID is available within the meta data of each VCF files. Figures 3.5 and 3.6 display the small portions of the metadata where the sample information and patient ID can be found. Applying the same methodology as that used for extracting the AD values, we were able to extract the patient IDs and include them onto the corresponding CSV files. The sample name, for both the normal and tumor samples, are hyphen-separated (e.g., TCGA-AB-2941-11A-01W-0745-08) with the patient ID number as the third value within this sequence. The subsequent information corresponds to the tier levels of the sub cut that was taken.



```
##SAMPLE=<ID=NORMAL,NAME=TCGA-AB-2935-11A-01W-0755-09,ALIQUOT_ID=d4bd29d6-e3ec-4007-82cf-10f0b87caaae,BAM_ID=53298512-bf55-49d7-8e0b-963853b9d347>
##SAMPLE=<ID=TUMOR,NAME=TCGA-AB-2935-03A-01W-0755-09,ALIQUOT_ID=a95d9260-b4d0-4b95-bfea-c605f940688c,BAM_ID=b6e2e2af-8c5a-4eb1-a572-08b1e3042f72>
```

Figure 3.5: Example of the sample information found on an AML VCF file.



ffe0c245-e88d-426c-aa44-f376dc6ab08c.wxs.MuSE.somatic_annotation.vcf

```
##SAMPLE=<ID=NORMAL,NAME=TARGET-10-PAPCVR-10A-01D,ALIQUOT_ID=0cd53c4f-10fd-53e3-b1e6-fc59a2cd0196,BAM_ID=052b069a-b4d7-4d50-aeab-62e499727f2d>
##SAMPLE=<ID=TUMOR,NAME=TARGET-10-PAPCVR-09A-01D,ALIQUOT_ID=6084a900-5bd7-57b4-9b3d-0e3621d9e5d1,BAM_ID=5d8c6c5e-7ad6-40f5-bd74-65b4c6ca788d>
```

Figure 3.6: Example of the same information found on an ALL VCF file.

The next step was to construct a binary matrix to indicate which variants occurred in which patients. Each row of this matrix represents a distinct variant found in the set of patients, and each column represents a unique patient. This matrix was constructed by writing a Python script to first compile a single list of unique SNVs contained in all the CSV files to create the rows, and then isolate the patient ID from the corresponding VCF and generate a list of unique patient entries. Once this was done, new columns were generated and named with the word “Patient_” followed by the ID entry. This resulted in each column representing a unique patient. For each column, the entries in the rows correspond to the listed unique variant, with a “0” indicating that this variant is not found in the patient, and a “1” indicating that the variant is present. The full Python script of this binary matrix construction process can be found in Appendix B.2 – B.4.

Organizing the data is crucial for any downstream analysis to performed, including the exploratory data analysis described in a later section of this chapter, as well as the future bioinformatics analyses proposed in Chapter 5. To do so, the columns that were extracted from the VCF files include:

Table 3.1: Variant file column names and descriptions.

Variant File Columns	
Column Name	**Description**
Chr	Chromsome number, or letter, in which the variant occurs on
ref_seq	The reference allele, or nonvariant base
var_seq1	First version of the variant sequence
var_seq2	Second version of the Variant sequence
alt_seq	The alternate allele, or variant base
VCF_ID	The downloaded Variant Call Format file name
Whole_Sample_ID	The patient biopsy sample identification number
Aliquot_ID	Extended version of sample ID, identifies the well within the plate in whice the sample was placed
Case_ID	Case identification number from the study or project

Additional information was extracted from the Whole_Sample_ID, which include the patient ID in the study and the main sample ID that was extracted from the patient. Ultimately this meant that each VCF file does not correspond to a unique patient but to a unique Whole_Sample_ID. In order to get an accurate count of the number of patients, only that partition of the Whole_Sample_ID containing the 4–6-character patient ID number was taken and inserted in an additional column labelled Patient_ID. Only unique values of column, ultimately meaning the unique patient counts, was found and then a column name “Patient_[unique patient ID]” was created for every patient. The rows values inserted in this each column were either a 1 or 0, identifying if the variant in the given row is found in each unique patient. Descriptions regarding the dimensions and content of the final version for the variant files can be found in the results sections, 4.3.1 and 4.3.2, below.

As both AML and ALL have been studied at the genomic level by many researchers, we have been able to utilize published articles to compile two lists of genes that have been reported to be associated with AML and ALL respectively, with specific focus on their genomic effects. The lists were organized on Excel which provided the capability of detecting any duplicated genes that had already been annotated. The columns on this list included the gene name and the reference in which it was mentioned. A third column was added later that correlated the reference number within this study.

3.2 Inclusion of Demographic Information

Once the master file in CSV format was created with the variant binary key, the next goal was to find the demographic details of the patients with these mutations. This step was successfully done using the cBioPortal website, using the original VCF file name, this entry matched with the another found on the data_clinical_patient.txt file for a TCGA patient study that was performed on the same samples collected in the prior section for the AML dataset and a corresponding TARGET patient study for the ALL dataset. Information provided on the files varied slightly between AML and ALL, it was found that the ALL had more demographic variables collected from the TARGET patient study than the ones found from the AML TCGA patient study. Also, for this step different databases for the same project, TCGA, on this portal were found leading to multiple demographics details for AML patients. The different databases for the AML TCGA project were PanCancer Atlas, Firehose, and New England Journal of Medicine. Although all three had overlapping variables in their database, they were all extracted, and a separate CSV master file was create containing only the unique variables. The Python script for that merges the three databases can be viewed in Appendix B.18. On this portal, ALL only had the one database for the TARGET study. Once all the demographic information was gathered here the information was then matched to the previous mutational data found from the VCF files using the unique patient ID (refer to Appendix B.17).

3.3 Exploratory Data Analysis

After gathering and organizing the data, the next step is to perform an exploratory data analysis and find certain trends and correlations within the AML and ALL patient groups, and then ultimately finding common trends between the AML and ALL groups. The first analysis was on the mutational data, reviewing this without any demographic information helps remove bias towards certain groups of people, age or gender.

Additionally, using the binary matrix generated from the previous step, the variant counts per patient were determined and a statical summary was performed on these values to its distribution. The last mutational data analysis that was performed was analyzing the chromosomal effects, by determining the number of variants per chromosome and then taking into consideration the size of these chromosome by divided the number of variants by the length of the chromosome [36]. The value of each chromosomal variant density was then ranked and graphed for further analysis (refer to Appendix C.2). Equation (1) is the formula for this calculation, where the individual chromosomes are represented by *chr*. It should be noted that within this formula the scale is being multiplied by 10^6, making the variant density per Megabase.

$$\text{density}(chr) = \frac{\text{Total number of variants on } chr}{\text{length}(chr)} * 10^6 \qquad (1)$$

For both the AML and ALL datasets, the mutational data of the normal and tumor samples were analyzed separately. Viewing the change type counts, was performed multiple ways, first method was taking the summation of the 12 individual counts and storing their values in a tabular format. Using the values, a column can be generated with the summation across each row representing the total number of variants for each ref base. Using this value, a conditional probability formula, Equation (3), was applied to each entry, where given that Y is the reference base being altered to any of the other three bases, X.

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)} = \frac{\text{Total count of } Y \rightarrow X \text{ mutations}}{\text{Total count } of\ Y \text{ mutations}} \tag{2}$$

To further this interpretation, refFlat and chromosome files were incorporated to expand the individual change types into their occurrences on the gene level. To briefly describe refFlat files, they are tab delimited text files, each file separated by chromosome, that contain all the unique genes along with their positional information and can be downloaded [37]. This information includes, transcription start and stop positions, coding region start and stop positions, and each exon within a coding region start and stop positions.

The positions are given in respect to the individual chromosomes, which is why the fasta files for each chromosome were assimilated into this study. Fasta files are formatted in a particular way, beginning with a header row that provides information on the sequence's identity, such as name, and in some cases certain identification numbers of the sequence. Within the next row, a ">" symbol is displayed, followed by the corresponding nucleotide or amino acid sequence.

Using Python, a script (refer to Appendix B.9) was created to find the unique genes within the mutational data. Once this list was compiled, it can then be cross referenced to the corresponding refFlat and chromosome fasta files, extracting each gene's positional information as well as the full reference genome transcription sequence (refer to Appendix B.10). With the full sequence, a count can be conducted for each nucleotide bases on the total number of times they occur within the unique gene sequences (refer to Appendix B.). Now taking the summation of these counts, it provides the total number that each base occurs in all the reference sequences (refer to Appendix B.11 – B.14).

Doing a summation on these values, an S_{xy}, can be calculated by taking the individual change type summations found previously and dividing it by the total number each bases occurrence in all the reference sequence. With this formula, it takes into consideration if a particular gene has a high normal occurrence of a particular base, its likelihood to be mutated is increased compared to a base that is less like to be found typically. The formula of S_{xy} is as follows:

$$S_{XY} = \frac{\text{Number of } X \rightarrow Y \text{ mutations observed in sample sequences}}{\text{Number of base } X \text{ observed in corresponding reference sequence}} \tag{3}$$

Based on the formula above, the necessary information to calculate this is the list of genes and their full sequence in which the mutations occur. With this a total count for each base in all the sequences can be found and then used as the denominator in this equation. The numerator is the change type that was previously calculated.

Now using the collected demographic data on only AML patients that had matches to the original VCF data, the patients without mutational data were not incorporated in this next analysis. By using the FAB M0 through M7 stages as a subgrouping method the first analysis performed was on the patients who were still living at that time and who were deceased against their overall survival (in months). The next step was then to determine the genders and ethnicities within these stages to get an idea of the patients within our mutational data (refer to Appendix C.3).

When working with categorical data, it is important to find methods that quantify data, one of which is FATHMM. By inputting all the SNVs, two separate list for the AML and ALL datasets, into FATHMM, a score is assigned to each as well as a determination of the harmfulness of the mutation. Using this information to understand further assess the genes that were found among the SNVs, a python script was created to isolate the unique genes while still maintaining the individual variant scores that were obtained. With the genomic information that was previous found on the genes for the other analyses, a formula was applied which generated an overall score for each gene, g.

$$P_l(g) = \frac{1}{\ln\ l(g)} \sum_v \text{F-score}(v) * [s(v, \text{tumor}) - s(v, \text{normal})] \qquad (4)$$

Equation (4) takes the summation of the products for each variant, *v*, score found from FATHMM, F-score, and the counts on which they occur in the normal and tumor samples, *s(v, *)*. This summation is then divided by the natural log of the length of the gene, *l(g)*. With this a every gene found within the mutational dataset can be assigned a quantitative value and sorted from highest to lowest. By using this scoring approach, the genes with the highest values can be considered most influential in the patients' development of AML and ALL.

Chapter 4: Results and Discussion

The main results of this work consist of four compiled data files and a statistical overview of them in preparation for more detailed bioinformatics and computational analyses to be conducted in the next two years. The four files consisted of mutational and clinical information on AML and ALL, allowing an exploratory statistical overview on them in parallel and a compare-and-contrast type of discussion of results.

4.1 Compiled Data Files

By compiling that data into files that contained similar information, the product of this was 4 separate files, containing AML and ALL variants and another that held the clinical information. These files are stored in the UTEP Bioinformatics data repository, and have the capability to be downloaded at the following link, https://datarepo.bioinformatics.utep.edu/getdata?acc=VYXBEYQ4OAFF5JR. The AML and ALL variant files, hold the binary matrix, with AML having 326 columns and 136,072 rows and ALL having 1,232 columns and 181,967 rows. The first row being the header for the columns, all other rows in these files represents a unique SNV that is found in both the normal and tumor samples. The first 28 columns in both files, are informational towards the unique SNV that they represent. The remaining columns are the patient binary matrix: in the AML file there are 149 columns that represent the patient's tumor samples and 149 columns for their normal samples, ALL has 603 columns for patient tumor samples and 603 for their normal samples. It should be noted, that after applying filter parameter when converting the ALL VCFs to CSV there were two patients that contained variants in the tumor sample but not the normal samples. These two patients are PARMEG and PASKRN, they were added to ensure that there were 603 patient columns for both normal and tumor samples. Within these columns, a 1 or 0 is entered, that determines if that unique

SNV is present in the patients corresponding sample (1 is found, 0 is not found). By doing so, this binary matrix allowed for a deeper look in the number of variants per patient.

The two clinical files, one for AML and one for ALL, do not hold the same consistency in information between each other as the variant files. This is due to separate projects that were conducted individually, TCGA for AML and TARGET for AML. However, they still contain important patient information regarding the mutational data. Within the clinical files, the AML has 56 columns and 150 rows, while ALL has 37 columns and 604 rows. In both files, the first row is the header for all the columns, while the remaining rows are the unique patients that have correlating information from the mutational data. Mentioned previously, the AML columns are from 3 separate sub studies. The first 6 columns are: the Patient ID assigned from the TCGA study, which is still used in all 3 sub studies, followed by variant counts for normal and tumor samples. For the remaining columns: 20 are from the New England Journal of Medicine, 23 are from Fire House, and 7 are from Pan Cancer. In the ALL clinical file, most of the columns are from the TARGET project, there are only 4 columns that were added regarding the variants occurrence in patients.

4.2 Statistical Overview of Demographic and Clinical Data for AML and ALL

The figures and tables below are using the FAB classification system, previously described, and comparing these classifications against the overall survival for patients living and deceased. The table below represents the number of patients broken up by the FAB classification system and some important demographic information.

Table 4.1: Categorized by the FAB classification system, the number of patients of within overall status, gender, and ethnicity demographic groups.

Classifications	Living	Deceased	Male	Female	White	Non-White
M0	5	11	12	4	11	5
M1	9	25	16	18	18	16
M2	10	23	14	19	21	12
M3	4	4	3	5	5	3
M4	11	23	20	14	27	7
M5	4	12	8	8	15	1
M6	0	3	3	0	2	1
M7	0	3	2	1	3	0
n.c.	1	1	2	0	1	1
Totals:	**44**	**105**	**80**	**69**	**103**	**46**

From this table we can interpret that the majority of the 149 patients in our AML dataset were classified as white males. Additionally of the 149 patients, only 44 patients' overall status is living, while 105 are deceased. You can also see the distribution of these demographics by the different classifications, such as the highest sample groups are the M1 and M4 classification with a total of 34 patients. M1 has an even distribution of gender and ethnicity, while M4 has more males, and the majority of these patients are white.

The figures below are notched boxplots, showing the overall survival in months for the patients who are deceased (Figure 4.1) or living (Figure 4.2) separated by their FAB classification.

With the black line representing the median overall survival, if this line surpasses the notches on the other classification boxes, then it can be concluded that there is significant difference with a 95% confidence interval.

With that, in figure 4.1, it can be said that the overall survival of patients with the M3 classification is significantly different than the other classification, excluding M0. Additional observations on this figure are the outliers that are found in the classifications M1, M2, and M4.

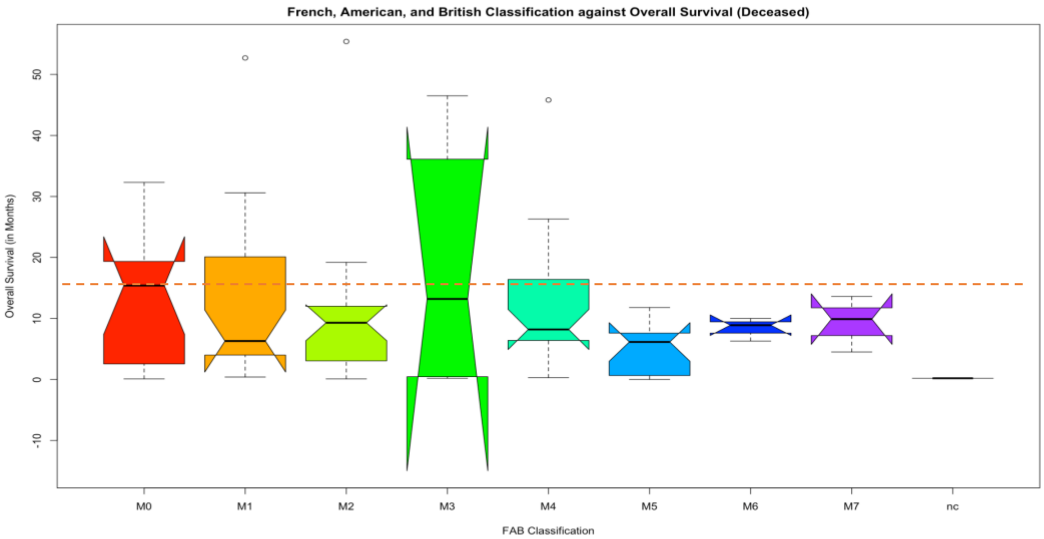


Figure 4.1: Notched boxplot of the FAB classification against the Overall Survival (in months) for deceased AML patients.

Looking at the next figure below, this is the same type of boxplot as the one above but is testing it for the patients who are still living. The main result here is that within this study there are no patients who are alive with type M6 and M7. Based on the median and its position to the other box plots, Figure 4.2, has the M3 classification being significantly different than only M1 and M2 classification with a 95% confidence interval.

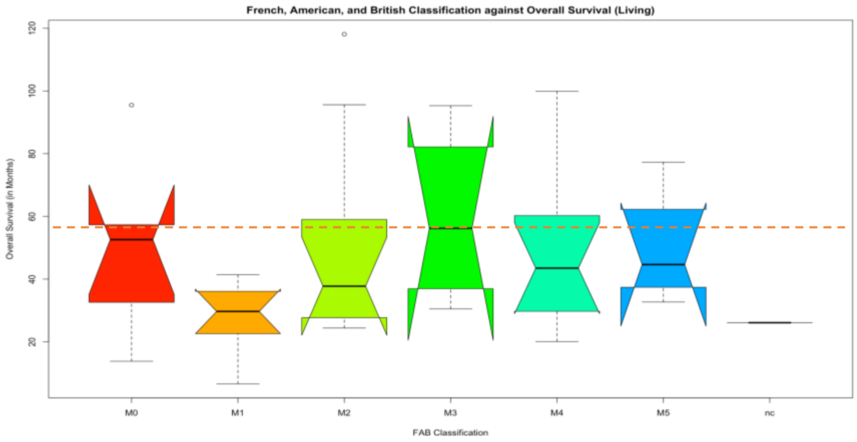


Figure 4.2: Notched boxplot of the FAB classification against the Overall Survival (in months) for living AML patients.

4.3 Statistical Overview of SNV Data for AML and ALL

4.3.1 SNV counts in AML dataset

For the 149 unique patients within the AML collection, the number of SNVs found in the normal sample was 6,769, whereas the tumor sample had 136,051 SNVs. A further analysis showed an overlap of these two lists of variants, which can be referred to as common. In the common area are a total of 6,749 SNVs that occur in both normal and tumor samples. Now using this value, the actual unique variants in both the normal and tumor groups are, 20 and 129,302 respectively.

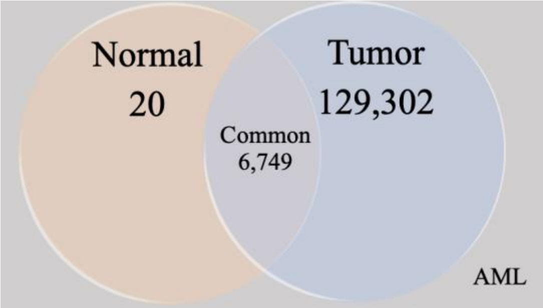


Figure 4.3: Venn diagram displaying the number of variants that are unique and shared between the normal and tumor samples in patients with AML.

Graphical representation of the number of variants per patient can be seen below, for both normal (orange line) and tumor (blue line) samples. The scatter plot graph has been organized to have the patient with the lowest tumor variant count first and then increasing to the patient with the highest tumor variant count last. Although two patients with the similar tumor variant counts, does not necessarily mean that they will have similar normal variants, Figure 4.4 actually shows that their normal variant counts can be drastically different. However, it is also seen on the extremities on this graph that the patient with the lowest and highest variant counts in their tumor samples also have the corresponding lowest and highest values in their normal samples. The five-point summary of these SNV counts is also shown in Table 4.2.

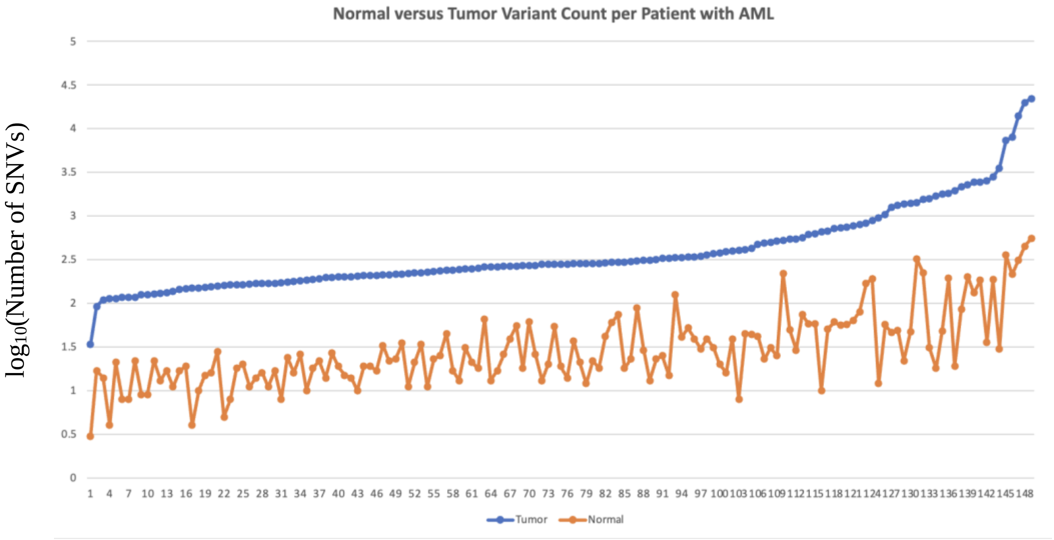


Figure 4.4: SNV counts (in $\log_{10}$ scale) in normal (orange) and tumor (blue) samples for patients with AML, sorted from lowest to highest tumor counts.

Table 4.2: Five-Point summary of the SNV counts for patients with AML.

AML Variant Count per Patient Five-Point Summary					
Sample Type	**Minimum**	**1st Quartile**	**Median**	**3rd Quartile**	**Maximum**
Normal	3	16	23	48	553
Tumor	34	197	282	547	22007

4.3.2 SNV counts in ALL dataset

For the total number of variants found in the normal samples there were 7,507, versus the tumor sample having 181,956. Among these, there were a total of 7,497 variants that are common between the normal and tumor samples. Using this value to exclude the overlap between the two original counts and find the unique variant counts between the normal and tumor, which are 10 and 174,459, respectively.

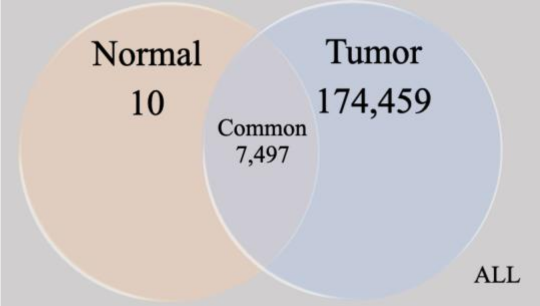


Figure 4.5: Venn diagram displaying the number of variants that are unique and shared between the normal and tumor samples in patients with AML.

Again, sorting the patients from smallest to largest tumor SNV counts (blue line) and then graphing along with their corresponding normal variants (orange line) resulted in Figure 4.6. It differs slightly from Figure 4.4 due to the patients who have variants found in the tumor sample, but none in the normal sample after filtering under the AD parameter when converting the VCF to CSV. For this reason, we can see a larger fluctuation in the normal variants per patient.

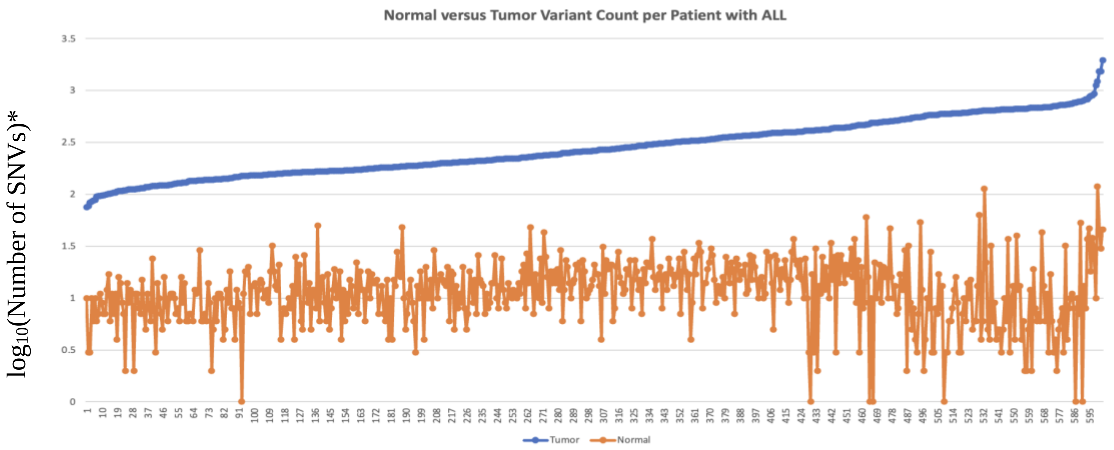


Figure 4.6: SNV counts (in log$_{10}$ scale) in normal (orange) and tumor (blue) samples for patients with ALL, sorted from lowest to highest tumor counts. *If SNV count = 0, we changed it to 1 in order to avoid taking log of 0.

The five-point summaries for the SNV counts in ALL normal and tumor samples are displayed in the Table 4.3.

Table 4.3: Five-Point Summary on the variant counts occurring in patients with ALL.

ALL Variant Count per Patient Five-Point Summary					
Sample Type	**Minimum**	**1st Quartile**	**Median**	**3rd Quartile**	**Maximum**
Normal	0	8	12	17	119
Tumor	75	169	264	441.5	1951

4.3.3 Comparing AML and ALL data

Referring to the tables of the five-point summaries above, the medians are quite similar, but the most notable difference found between the two leukemia's is the minimum and maximum values. AML had a much larger range between these two values, at 21,973, whereas ALL was much closer together at 1,876. On a closer look at the SNV counts of individual patients, we found that this can be explained by the presence of outliers. Actually only 13 (< 9%) of the patients with AML have tumor counts higher than the maximum number found in the ALL tumor sample type. As yet, it would not be possible to conclude that there is any significant difference between the ranges of tumor sample SNV counts between AML and ALL.

4.3.4 AML mutation levels

For the AML SNV dataset, counts for the 12 possible change types in the normal and tumor samples are presented in Tables 4.4 and 4.5. The left-most column lists the reference allele (Ref) while the top row lists the alternate allele (Alt).

Table 4.4: AML Normal Change Type Counts.

Ref/Alt	**A**	**G**	**T**	**C**	**Sum**
A	0	901	212	725	1838
G	739	0	862	474	2075
T	197	850	0	859	1906
C	850	480	790	0	2120

Table 4.5: AML Tumor Change Type Counts.

Ref/Alt	**A**	**G**	**T**	**C**	**Sum**
A	0	10896	11394	2669	24959
G	20523	0	31580	4270	56373
T	9918	2853	0	11160	23931
C	31541	4330	19861	0	55732

The key difference between Tables 4.4 and 4.5 is that the SNV counts are significantly higher in the tumor samples than in the normal samples. Additionally, just interpreting the individual counts among the 12 change types, the highest change in the normal sample is the nucleotide change from A to G, versus its tumor sample counterpart, having the highest nucleotide change from G to T. The last column on both tables (Sum) is the overall sum of the Ref being changed. This revealed a huge disparity between the normal and tumor samples as well. Although the individual counts within the given Ref row differ, one can see that there is less variability in the sums of the normal samples. This is not the case in the tumor samples, the Ref sum are drastically different, showing that the G and C ref alleles having double the number of alterations as the T and A nucleotides.

The results of the individual change type conditional probabilities for normal and tumor are on Tables 4.6. and 4.7, respectively. With the reference allele in the first vertical column and the alternate allele as the first horizontal row.

Table 4.6: AML Normal Conditional Probability.

Ref/Alt	**A**	**G**	**T**	**C**
A	0	0.4902	0.1153	0.3945
G	0.3562	0	0.4154	0.2284
T	0.1033	0.4460	0	0.4507
C	0.4009	0.2264	0.3726	0

Table 4.7: AML Tumor Conditional Probability.

Ref/Alt	**A**	**G**	**T**	**C**
A	0	0.4366	0.4565	0.1069
G	0.3641	0	0.5602	0.0757
T	0.4144	0.1192	0	0.4664
C	0.5659	0.0777	0.3564	0

Using the normal sample conditionally probability as a baseline of the acceptable changes that can occur and comparing these against the tumor samples. There is a visible shift in the probabilities for each row. First looking at the A reference allele, the lowest probability in the normal sample is the nucleotide change from A to T, whereas the tumor sample has this same change as the highest. Another difference found in these tables is the change from T to A, the normal sample has this change as the lowest probability, but the tumor sample has shifted into, a close, second highest probability.

Graphical 3D bar plot representation of Tables 4.4 and 4.5 can be seen below for the AML normal and tumor change types. The most notable difference between the normal and tumor graphs is at the vertical-axis scaling, the highest value among 12 change types is 901, whereas in the tumor sample it is 31,580. Other than the drastically increased scaling factor from the normal to tumor counts, there are also visible changes in the change-type distributions that were assessed in the conditional probabilities on Tables 4.6 and 4.7.

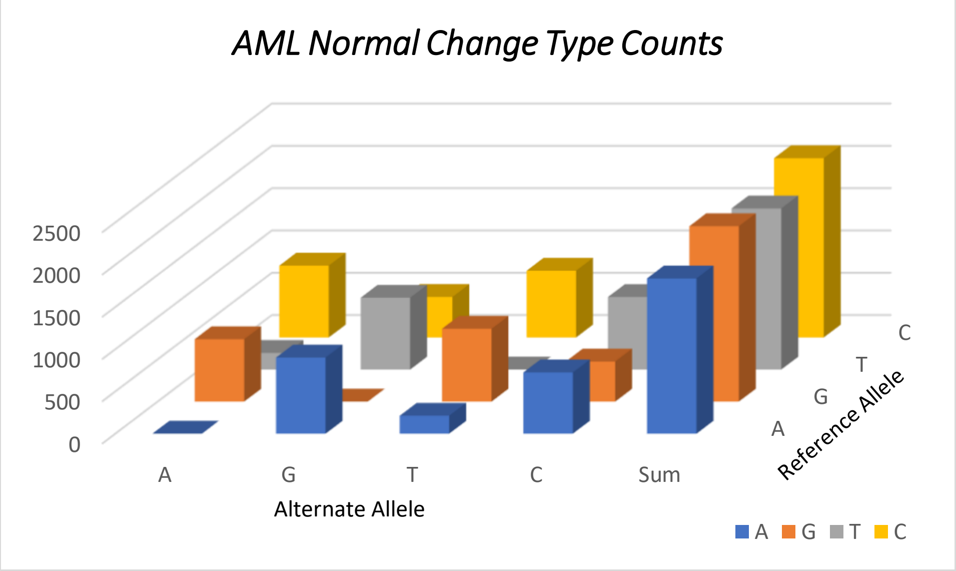


Figure 4.7: Couns of individual change types found in the normal samples of patients with AML.

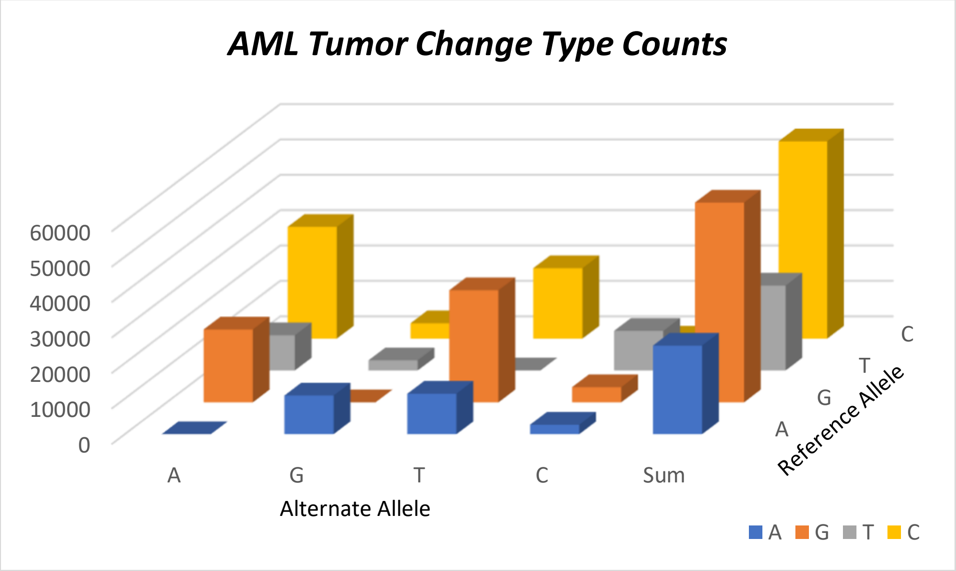


Figure 4.8: Counts of individual change types found in the tumor samples of patients with AML.

Figure 4.9 displays the AML variant densities in different chromosomes, with the normal sample assessment on the left and the tumor sample on the right.

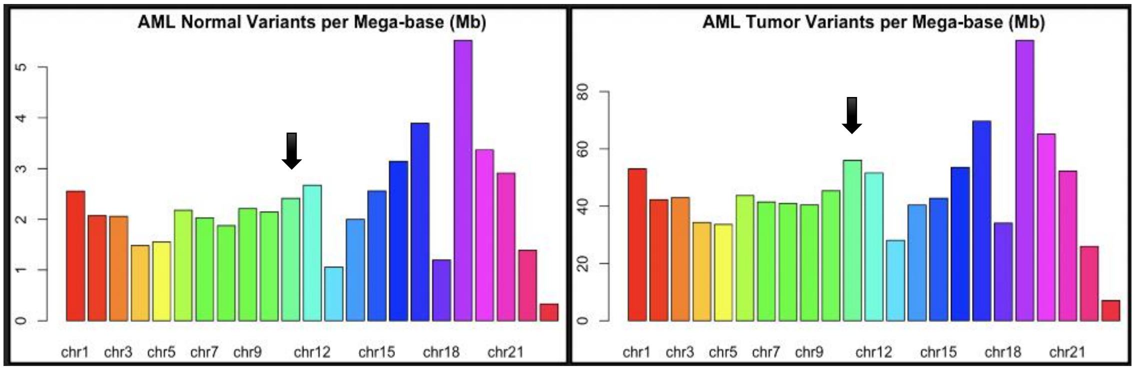


Figure 4.9: Bar graphs (normal samples on left, tumor samples on right) displaying the variant densities per chromosome in patients with AML.

Similar to the change type counts, the scale disparity between the normal and tumor samples is very noticeable. Also, when looking at the distribution trend of the bar graph, it can be observed in the ledge disparity, or shifts in the individual bars, of this distribution that chromosome 11 has an increase when comparing its neighboring chromosome on the normal and tumor samples.

Tables 4.8 and 4.9 provide another way to view the AML chromosome bar graph. These provide the numerical values for the normal and tumor variants per megabase after sorting and ranking them. Table 4.8 shows the chromosomes ranks when sorted based on the Normal Variants per Megabase column, while Table 4.9 is ranked after being sorted based on the Tumor Variant per Megabase columns. After sorting under these conditions, we can numerically see what the bar plot above had displayed, that chromosome 11 does jump to a higher rank going from the 9th highest in normal samples to the 4th highest in tumor samples.

Also included in the tables is a column called Ratio, whose entries are the ratio between tumor and normal calculated by taking the Tumor value and dividing it by Normal. These values allow for further interpretation to be done on chromosomes that appear to have higher than average ratio but may not have shifted after sorting. There are no biologically significant outliers present in the AML dataset, although the highest ratio found was on Chromosome 18.

Table 4.8: Chromosome variant density per Megabase on AML patients, sorted on the Normal Variants per Megabase column from largest to smallest values and their given rank in the first column.

Rank	Chromosome	Tumor	Normal	Ratio (Tumor/Normal)
1	chr19	121.50	6.65	18.26
2	chr17	83.20	4.38	18.98
3	chr16	64.11	3.90	16.45
4	chr20	77.46	3.82	20.29
5	chr21	62.43	3.32	18.81
6	chr12	61.29	3.22	19.04
7	chr15	51.91	3.14	16.54
8	chr1	62.83	2.98	21.05
9	chr11	64.73	2.85	22.71
10	chr6	51.56	2.56	20.11
11	chr9	47.57	2.48	19.20
12	chr10	52.80	2.44	21.60
13	chr2	50.06	2.43	20.58
14	chr14	48.06	2.42	19.86
15	chr3	50.21	2.38	21.14
16	chr7	48.69	2.31	21.08
17	chr8	47.19	2.26	20.88
18	chr5	39.39	1.84	21.41
19	chrX	32.70	1.74	18.76
20	chr4	39.87	1.68	23.77
21	chr18	38.92	1.36	28.70
22	chr13	32.60	1.28	25.53
23	chrY	8.28	0.33	24.95

Table 4.9: Chromosome variant density per Megabase on AML patients, sorted on the Tumor Variants per Megabase column from largest to smallest values and their given rank in the first column.

Rank	Chromosome	Tumor	Normal	Ratio (Tumor/Normal)
1	chr19	121.50	6.65	18.26
2	chr17	83.20	4.38	18.98
3	chr20	77.46	3.82	20.29
4	chr11	64.73	2.85	22.71
5	chr16	64.11	3.90	16.45
6	chr1	62.83	2.98	21.05
7	chr21	62.43	3.32	18.81
8	chr12	61.29	3.22	19.04
9	chr10	52.80	2.44	21.60
10	chr15	51.91	3.14	16.54
11	chr6	51.56	2.56	20.11
12	chr3	50.21	2.38	21.14
13	chr2	50.06	2.43	20.58
14	chr7	48.69	2.31	21.08
15	chr14	48.06	2.42	19.86
16	chr9	47.57	2.48	19.20
17	chr8	47.19	2.26	20.88
18	chr4	39.87	1.68	23.77
19	chr5	39.39	1.84	21.41
20	chr18	38.92	1.36	28.70
21	chrX	32.70	1.74	18.76
22	chr13	32.60	1.28	25.53
23	chrY	8.28	0.33	24.95

4.3.5 ALL mutational levels

With a similar approach, utilizing the mutational data from GDC on patients with ALL, the SNV experiment was executed. Focusing on the same trends in which the individual change types occur, it is important to annotate the data imbalance. Comparing against the AML, disregarding the previous biological factor being that the SNV arises in a different cell type, the ALL dataset had 3 times as many patients.

A comparison between the normal and tumor samples change type counts can be seen on the tables below, Tables 4.10 – 4.13. From these tables one distinction that can be made is in the summation column, when looking at the frequency that the G nucleotide changes in the normal samples is the lowest occurring change, while in tumor samples it is shown to be the second highest. Additionally, when looking at the alternate allele and what G is changing to, the most common appear to be A and T. This change can alter the GC content of gene and ultimately changing the thermal stability. Essentially, the higher the GC content, the more stable the double stranded helical molecule is [12].

Table 4.10: ALL Normal Change Type Counts.

Ref/Alt	**A**	**G**	**T**	**C**	**Sum**
A	0	991	210	830	2031
G	1065	0	578	345	1988
T	167	1000	0	1005	2172
C	849	321	1208	0	2378

Table 4.11: ALL Tumor Change Type Counts.

Ref/Alt	**A**	**G**	**T**	**C**	**Sum**
A	0	16412	5139	3941	25492
G	27922	0	16822	6660	51404
T	5244	4439	0	16649	26332
C	67431	6661	25681	0	99773

Table 4.12: ALL Normal Conditional Probabilities.

Ref/Alt	A	G	T	C
A	0	0.4879	0.1034	0.4087
G	0.5357	0	0.2908	0.1735
T	0.0769	0.4604	0	0.4627
C	0.3570	0.1350	0.5080	0

Table 4.13: ALL Tumor Conditional Probabilities.

Ref/Alt	A	G	T	C
A	0	0.6438	0.2016	0.1546
G	0.5432	0	0.3272	0.1296
T	0.1991	0.1686	0	0.6323
C	0.6758	0.0668	0.2574	0

Using the previous tables information, the conditional probabilities were calculated as an alternative way of viewing the change types between the normal and tumor ALL samples. When comparing the normal conditional probability table to the tumor conditional probability table, using the previous method considering the normal conditional probabilities as a baseline for acceptable changes given what base is being mutated. For this reason, the analysis is conducted row by row for shifts in the baseline found from the normal samples. When considering that reference A allele is being mutated, the conditional probability that the alternate allele changes to a T is almost double from normal to tumor. Additionally, the reference allele T changing to the alternate allele G or C are both approximately 0.46 in normal samples, however when comparing this to the tumor probabilities these values shift to 0.17 and 0.63, respectively. Finally, given that the mutation is on the reference allele C, similar to the reference allele A changing to a T, the alternative allele being A in this case doubles from the normal to tumor samples.

The figures below are the 3D bar plot of the ALL normal and tumor 12 change types counts. Here we can see the scaling being drastically higher in the tumor samples than the normal, which is to be expected. For this reason, the graphs are displayed separately to avoid loss of visual effect when using the same scaling criteria. The graph is also confirming the results found in the conditional probability and the shifts that were interpreted. The most significant result is the dramatic increase found from the C reference allele mutating to an A when comparing the normal to the tumor.

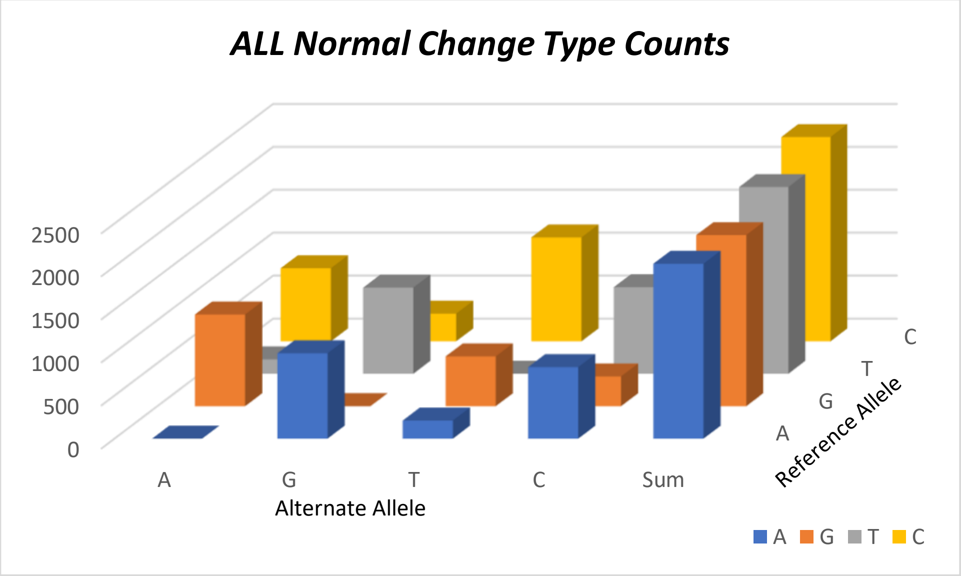


Figure 4.10: 3-D Bar graph of the individual change types found in the normal samples of patients with ALL.

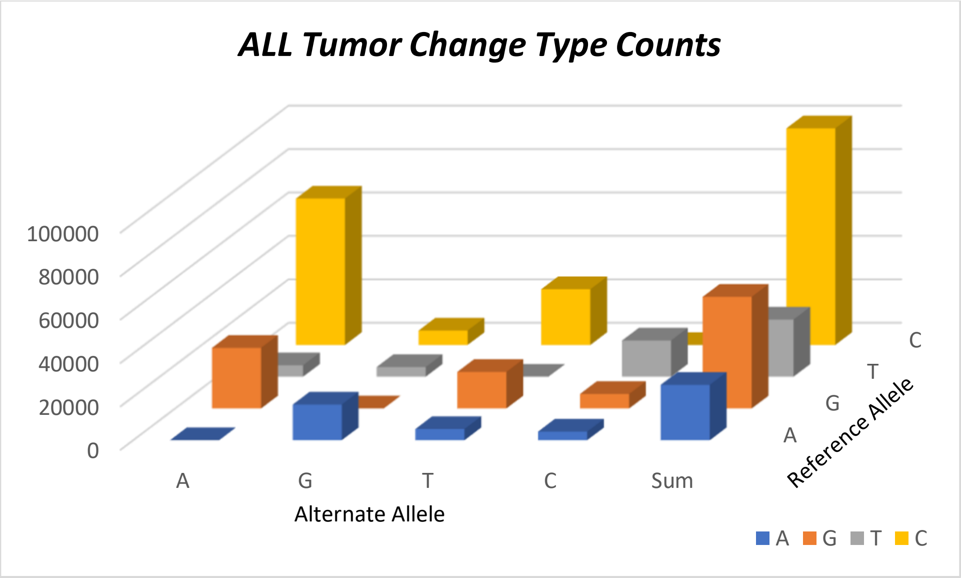


Figure 4.11: 3-D Bar graph of the individual change types found in the tumor samples of patients with ALL.

Furthermore, when analyzing the variants per megabase on each chromosome for the ALL dataset, the scaling when looking at the normal versus tumor samples is drastically higher. Using the same method as the AML in assessing the ledge disparities among the chromosomes, visually it shows subtle changes among all the chromosomes, but the most significant change is chr2 drastic increase when going from the normal to tumor graph.

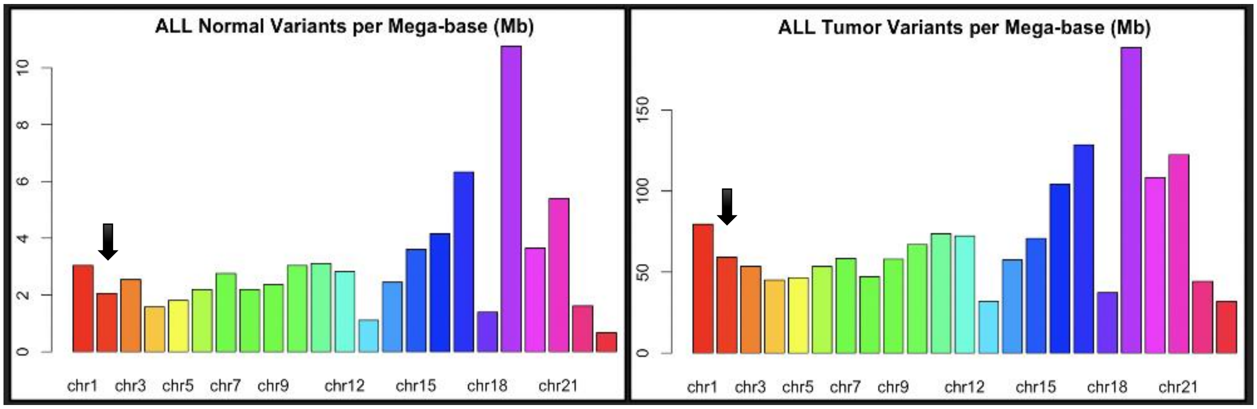


Figure 4.12: Bar graphs (normal samples on left, tumor samples on right) displaying the variant densities per chromosome in patients with ALL.

The table below corresponds to bar graph above with the chromosome variants density numerical values. The values in each table are the same, however the sorting method is altered between the two. In the first table the Normal Variants per Megabase column is sorted based on largest to smallest, whereas the second table is being sorted by the Tumor Variants per Megabase column. The ranking is listed to help assess how the chromosomes are being shifted allowing it for an alternative reading format than the bar graph.

Table 4.14: Chromosome variant density per megabase on ALL patients, sorted on the Normal Variants per Megabase column from largest to smallest values and their given rank in the first column.

Rank	Chromosome	Tumor	Normal	Ratio (Tumor/Normal)
1	chr19	188.73	10.76	17.53
2	chr17	128.61	6.33	20.32
3	chr21	122.39	5.39	22.69
4	chr16	104.27	4.15	25.12
5	chr20	108.34	3.65	29.71
6	chr15	70.80	3.61	19.62
7	chr11	73.66	3.11	23.69
8	chr1	79.56	3.04	26.20
9	chr10	66.94	3.03	22.06
10	chr12	72.35	2.83	25.58
11	chr7	58.43	2.76	21.21
12	chr3	53.60	2.54	21.09
13	chr14	57.39	2.45	23.45
14	chr9	57.99	2.36	24.54
15	chr6	53.36	2.19	24.37
16	chr8	47.16	2.19	21.53
17	chr2	59.13	2.05	28.81
18	chr5	46.51	1.81	25.74
19	chrX	44.24	1.61	27.40
20	chr4	45.04	1.58	28.56
21	chr18	37.33	1.39	26.79
22	chr13	31.94	1.11	28.76
23	chrY	31.87	0.66	48.00

Comparing the two tables above and below, there are the shift in ranks among the chromosomes that we found in the previous bar plot. When focusing on chr2, this is the change that was found and now can be confirmed using this ranking method. In the table above, representing the normal sample, chr2 is ranked at 17th, now checking its rank in the tumor sample, chr2 is ranked at 11th.

Alternatively, reviewing the ALL Ratio column entries, comparing with the AML, there is a slightly higher ratio average, but also within the ALL dataset there is a biologically significant outlier found on Chromosome Y. Previous work has also confirmed the significance associated to variants found in DQA1, supporting the ALL risk in males [51]. Although there was no change in rank when analyzing after sorting, this value shows a much higher than average mutation ratio when comparing tumor and normal.

Table 4.15: Chromosome variant density per megabase on ALL patients, sorted on the Tumor Variants per megabase column from largest to smallest values and their given rank in the first column.

Rank	Chromosome	Tumor	Normal	Ratio (Tumor/Normal)
1	chr19	188.73	10.76	17.53
2	chr17	128.61	6.33	20.32
3	chr21	122.39	5.39	22.69
4	chr20	108.34	3.65	29.71
5	chr16	104.27	4.15	25.12
6	chr1	79.56	3.04	26.20
7	chr11	73.66	3.11	23.69
8	chr12	72.35	2.83	25.58
9	chr15	70.80	3.61	19.62
10	chr10	66.94	3.03	22.06
11	chr2	59.13	2.05	28.81
12	chr7	58.43	2.76	21.21
13	chr9	57.99	2.36	24.54
14	chr14	57.39	2.45	23.45
15	chr3	53.60	2.54	21.09
16	chr6	53.36	2.19	24.37
17	chr8	47.16	2.19	21.53
18	chr5	46.51	1.81	25.74
19	chr4	45.04	1.58	28.56
20	chrX	44.24	1.61	27.40
21	chr18	37.33	1.39	26.79
22	chr13	31.94	1.11	28.76
23	chrY	31.87	0.66	48.00

4.3.6 Mutation levels in AML and ALL

The formula for S_{xy} in equation (3) of Section 3.3 was used to calculate the mutation level of each change type of a base taking into account its frequency of occurrence in the genome sequences. Tables 4.16 and 4.17 respectively display the AML and ALL S_{xy} values for each change type. The values in the last column is the summation of each reference base count for all of the genes containing SNVs in our dataset.

Table 4.16: S_{xy} results, based on Equation (3), for patients with AML, entries in columns A,G,T, and C are multiplied by a scale of 10^6.

Ref/Alt	A	G	T	C	Ref Base Count
A	0.00	31.0	32.5	7.6	351106379
G	77.0	0.00	118.5	16.0	266526958
T	25.9	7.5	0.00	29.2	382805654
C	123.4	16.9	77.7	0.00	255650983

Table 4.17: S_{xy} results, based on Equation (3), for patients with ALL, entries in columns A,G,T, and C are multiplied by a scale of 10^6.

Ref/Alt	A	G	T	C	Ref Base Count
A	0.00	47.1	14.8	11.3	348148321
G	104.6	0.00	63.0	24.9	266991806
T	13.8	11.7	0.00	43.8	380101844
C	263.4	26.0	100.3	0.00	255992249

For both AML and ALL, the highest S_{xy} value is the change type from C to A. In AML, that next highest was G to T, followed by C to T and G to A. In ALL, the order slightly varied with G to A as the second highest, then C to T, and finally G to T. Overall the displayed S_{xy} values suggest that the SNV's favor mutations that decrease the DNA GC content in these patients. Additionally, the "Ref Base Count" columns in these tables also show, for both AML and ALL, that both G and C occurrences in the reference sequences of the mutated genes are much lower than both A and T.

4.4 Summary of Main Results

In both AML and ALL, using the normal samples as a baseline of the acceptable mutations that are found in patients with leukemia, there are noticeable shifts in the data that identify favoritism of the SNVs found in tumor samples. To address the change types patterns displayed, there is a large increase in the mutations that occur on original reference bases G and C nucleotide. Other trends in that the change types showed that the conditional probability of these changes mutating to alternate bases A and T nucleotides. This reveals a decrease in the GC content of DNA, which has shown to affect the structural integrity of proteins.

To further support this biologically significant finding, the S_{xy} mutational values were conducted. Recalling that these values incorporate the reference base counts, and in doing so, this alternate method confirmed the favored change types from reference bases G and C altering to either A and T.

Continuing with the mutational data, chromosome frequencies were also determined. After taking into consideration the chromosomes length and calculating a density of the SNV occurrences, in both AML and ALL changes were noticed, however there was the greatest change displaying an increase on chromosome 11 for AML and chromosome 2 in ALL.

Using the demographic data, the main discovery found was when comparing the staging (M0-M7) to the overall survival of patients (living or deceased) who have been diagnosed with AML. It was found that patients with M6 or M7 have all deceased, thus assuming that this particular stage of AML result in a very poor prognosis for these patients.

Chapter 5: Conclusion & Future Work

This study has been more about the journey than the destination, sparking ideas to be explored further by applying computational approaches to analyze the organized merged clinical and genomics data for AML and ALL. After making a brief conclusion in Section 5.1 based on results obtained to date, extensions from my current work, planned to be completed within the next two years, are outlined in Section 5.2.

5.1 CONCLUSION

The process of utilizing multiple public portals to extract, compile, and organize datasets can be most strenuous due to vast amount of information collected, but having the properly merged datasets is key to downstream analysis that can link patients' genetic variant profiles with their demographic and clinical information in relation to their survival outcomes. This step provided the necessary materials for an initial exploratory overview that showed promising observable trends for detailed investigation in the future. In addition, it has also served as a guide that suggests suitable bioinformatics analyses to be conducted and the software tools to be implemented for such investigations.

5.2 FUTURE WORK

The main objective of the proposed future work would be to expand the information that has been gathered and implement more refined methodology for detailed analyses of the AML and ALL datasets. The specific aims are:

1. Develop a new scoring scheme to assess how likely a protein-coding gene is associated with AML and ALL The score will be based on the functional effects of the individual SNVs contained within the genes and used to provide a ranking. The top-ranking genes will then be

selected for downstream bioinformatics studies with special attention paid to their connections with phosphatases, kinases, and glycosyltransferases.

2. Analyze the patients' survival status and overall survival time with relevant clinical and demographic variables (e.g., FAB classification in AML, CNS stage in ALL) plus the selected top-ranking genes, along with the known AML- and ALL-associated genes compiled and listed in Appendix A. Multiple linear and logistic regression methods will be applied.

Proposed approaches to achieve the above aims are described below.

5.2.1 Variant Scoring Scheme for Identifying Likely AML- and ALL-associated Genes

Step 1. Software tools for predicting functional effects of individual SNVs

To assist with analyzing single nucleotide variants found, a number of existing software tools are available. After an initial survey of them to assess their usability, we settled on using the following:

(i) FATHMM ([38], https://fathmm.biocompute.org.uk/fathmm-xf/). This is an online tool that predicts the pathogenic behavior of an SNV, allowing the user to adjust parameters. For every prediction when using this server, a p-value is assigned to test the significance. Other studies have shown that FATHMM to be most efficient against its competitors [11]. The original FATHMM scores that were calculated and interpreted in the results section were performed using the webserver. The input file was coded using python, a job was created in the queue, and the output file was downloaded once completed. This process has an alternative method, by using a downloadable package to be created on a personal device. This would eliminate the mandatory queue waitlist entirely and make utilizing this software more efficient.

(ii) PROVEAN [39]. this software is used to predict the variants effect. Utilizing the Oncominer tool to calculate PROVEAN scores for the nonsynonymous variants is also going to be performed. To be able to run this software missing components will need to be identified for each variant, such as the amino acid change in a specific format and a reference number both of which can be found in the original VCF files. This information was found much later when originally working on this research project, but the complication that arose is extracting the information from the column in which it is found. To extract the information column, the original python code that converts a VCF file into csv will need to be modified to include this as a column on the output. Next this column varies depending on the variant and includes extra information that is unnecessary for this software. Essentially the entire column will have to be cleaned to create two columns with the correct format of the amino acid alteration and the reference sequence. The format is displayed as follows: the original amino acid (Single capital letter abbreviation), the amino acid sequence position, and finally the new amino acid (single capital letter abbreviation). These three values have no spaces or special characters between them. The reference sequence is the second column that gives an identification number of the gene sequence. This number can be read into the Oncominer pipeline, in which it then will refer to the original sequence being modified and return a prediction score based on the change that was previously mentioned.

(iii) STRUM [45]. Other approaches in interpreting SNV, besides the deterioration in genomic function is the protein folding changes that can occur when mutation arises. For this a software online called STRUM can be used. This method is used for nonsynomous mutations and their position to predict how they change the folding on that protein molecule. As small as these changes may seem, just a single nucleotide change can alter the amino acid altering the hydrophobicity, acidity, and over shape of the protein. When exploring the STRUM's capabilities, it was found to be very time consuming but returns interesting results. As an experiment, a single amino acid change was used in less than a 15-length protein strand, this program took an estimated 5 days to return output versus the same day results the previously mentioned tools took. As promised, it returned the delta delta G energy change from the original and mutated strand, as well as a 3D models that displayed the bonding changes that occurred. The future intention for STRUM is to utilize this program without having to submit a job on the webservers queue. The main disadvantage with this software was the highly inefficient runtime, it is expected that if this software can be run on a personal device or an HPC, essentially avoiding the webserver, will help with this time issue. The main website has a package that can be downloaded, however if this method does not perform as expected, the next step would be to take a very select group of SNVs. This group would be formed using the other softwares and have provided evidence of the SNVs contribution in patients with acute leukemia.

(iv) SNPnexus ([40-44], https://www.snp-nexus.org/v4/). This tool has undergone recent updates in the past decade that has made it a more reliable source in predictions on the functionality consequences that point mutations can cause. With the capabilities to choose from with Human Genome Reference, it generates hits on the imported data. This web-based server provides information on the genomic coordinated based on their physical and cytogenetic positions. Additionally, this site also provides users with known SNV overlaps and if available a link to dbSNP, as well as the closest gene, the type of gene, and a predicted consequence.

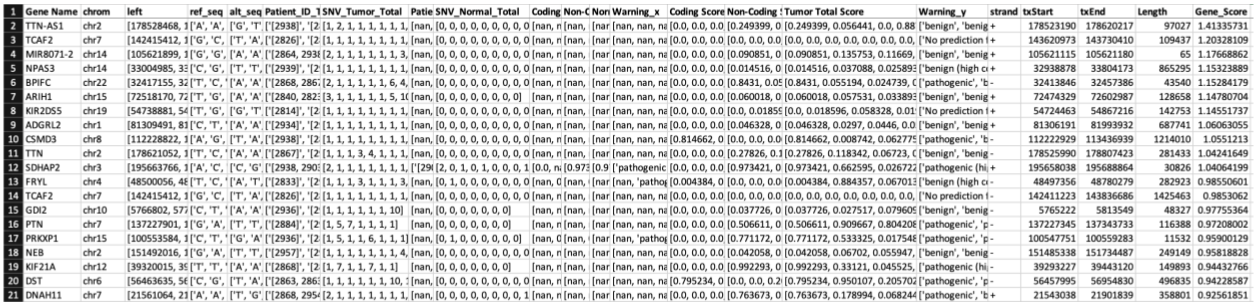
Other members of our group had previously used FATHMM and PROVEAN to predict the functional effects of individual variants on protein-coding genes [50]. Combining these with the variants' occurrence frequencies among patients in the dataset resulted in a scoring scheme that helped predict several likely genes related to prostate cancer. I will first try using the same method on our leukemia SNV datasets to get a baseline prediction performance. Then I will incorporate the STRUM and SNPnexus results to form a new scoring scheme to improve prediction performance.

Furthermore, incorporate additional prediction software that validate the importance of particular SNVs and the genes found. Some examples of this would be PROVEAN and STRUM, but also contribute to the Oncominer pipeline with a FATHMM feature that could be used for researchers. Additionally, this process can be implemented in the Oncominer pipeline with a few other modifications to the code. Allowing the user to use a default selection of variants (every variant within a VCF, or OMI file) or be able to specify only certain variants based on their classification and score (i.e. nonsynonymous and/or pathogenic).

Step 2. Identification of high-scoring genes for downstream bioinformatics analysis

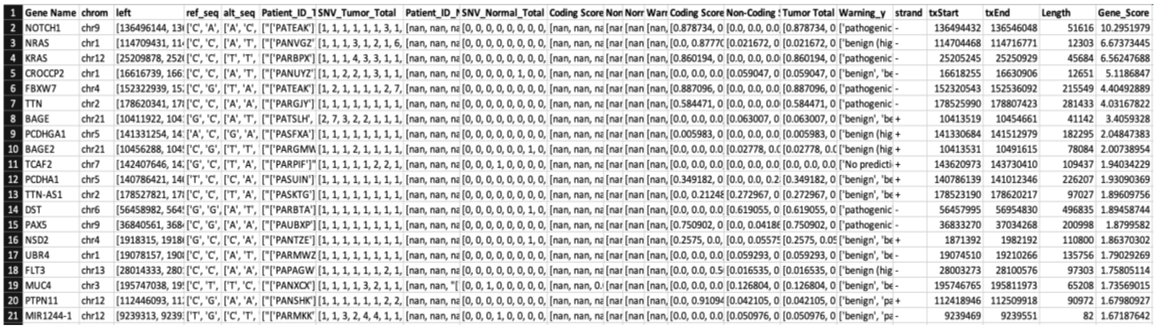
After utilizing the functional effect evaluation tools above on the individual variants, the next step of this investigation would be to score the protein-coding genes based on the functional effects of the SNVs contained in them and identify the high-scoring gene. As a preliminary trial, I have attempted this gene scoring process using the FATHMM tool alone. FATHMM scores for the unique set of genes found amongst the SNV lists containing normal and tumor variants. The figures below show the top 20 genes with highest scores for AML and ALL respectively based on their score from the gene and variant information formula from Equation (4).

Within the first column there is the gene name and the last column is the score that was found. The warning column is another categorical variable that FATHMM has assigned to each variant that identifies its harmfulness, ranging from benign (not harmful) to pathogenic (harmful), in some cases no predictions can be found.



1	Gene Name	chrom	left	ref_seq	alt_seq	Patient_ID_T	SNV_Tumor_Total	Patie	SNV_Normal_Total	Coding	Non-C	Non	Warning_x	Coding Score	Non-Coding S	Tumor Total Score	Warning_y	strand	txStart	txEnd	Length	Gene_Score
2	TTN-AS1	chr2	[178528468, 1	['A', 'A',	['G', 'T'	['[2938]', '[2	[1, 2, 1, 1, 1, 1, 1, 1,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.249399, 0	[0.249399, 0.056441, 0.0, 0.88	['benign', 'benig	+	178523190	178620217	97027	1.41335731
3	TCAF2	chr7	[142415412, 1	['G', 'C',	['T', 'A',	['[2826]', '[2	[1, 1, 1, 1, 1, 1, 1, 1,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.0, 0.0, 0.0	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,	['No prediction f	+	143620973	143730410	109437	1.20328109
4	MIR8071-2	chr14	[105621899, 1	['G', 'G',	['A', 'A'	['[2864, 2938	[2, 1, 1, 1, 1, 1, 1, 3,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.090851, 0	[0.090851, 0.135753, 0.11669,	['benign', 'benig	+	105621115	105621180	65	1.17668862
5	NPAS3	chr14	[33004985, 33	['C', 'G',	['T', 'T',	['[2939]', '[2	[1, 1, 1, 1, 1, 1, 1, 1,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.014516, 0	[0.014516, 0.037088, 0.025893	['benign (high c	+	32938878	33804173	865295	1.15323889
6	BPIFC	chr22	[32417155, 32	['T', 'C', '	['A', 'A'	['[2868, 2867	[2, 1, 1, 1, 1, 1, 6, 4,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.8431, 0.05	[0.8431, 0.055194, 0.024739, 0	['pathogenic', 'b	-	32413846	32457386	43540	1.15284179
7	ARIH1	chr15	[72518170, 72	['T', 'G',	['A', 'A'	['[2840, 2823	[3, 1, 1, 1, 1, 1, 5, 10	[nan,	[0, 0, 0, 0, 0, 0, 0, 0]	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.060018, 0	[0.060018, 0.057531, 0.033893	['benign', 'benig	+	72474329	72602987	128658	1.14780704
8	KIR2DS5	chr19	[54738881, 54	['T', 'G',	['G', 'T'	['[2814]', '[2	[1, 1, 1, 1, 1, 1, 1, 1,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.0, 0.01859	[0.0, 0.018596, 0.058328, 0.01	['No prediction f	+	54724463	54867216	142753	1.14551737
9	ADGRL2	chr1	[81309491, 81	['C', 'T', '	['A', 'A'	['[2934]', '[2	[1, 1, 1, 1, 1, 1, 1, 1,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.046328, 0	[0.046328, 0.0297, 0.0446, 0.0	['benign', 'benig	+	81306191	81993932	687741	1.06063055
10	CSMD3	chr8	[112228822, 1	['A', 'G',	['T', 'A',	['[2938]', '[2	[1, 1, 1, 1, 1, 1, 1, 1,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.814662, 0	[0.0, 0.0, 0.0	[0.814662, 0.008742, 0.062775	['pathogenic', 'b	-	112222929	113436939	1214010	1.0551213
11	TTN	chr2	[178621052, 1	['T', 'C', '	['A', 'A'	['[2867]', '[2	[1, 1, 1, 3, 4, 1, 1, 1,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.27826, 0.1	[0.27826, 0.118342, 0.06723, 0	['benign', 'benig	-	178525990	178807423	281433	1.04241649
12	SDHAP2	chr3	[195663766, 1	['A', 'C',	['C', 'G'	['[2938, 2903	[2, 1, 1, 1, 1, 1, 1, 1,	['[29	[2, 0, 1, 0, 1, 0, 0, 0, 1	[0.0, na	[0.973	[0.9	['pathogenic	[0.0, 0.0, 0.0	[0.973421, 0	[0.973421, 0.662595, 0.026722	['pathogenic (hi	+	195658038	195688864	30826	1.04064199
13	FRYL	chr4	[48500056, 48	['T', 'C', '	['A', 'T',	['[2833]', '[2	[1, 1, 1, 3, 1, 1, 1, 3,	[nan,	[0, 1, 0, 0, 0, 0, 0, 0, 0	[nan, 0	[nan, '	[nar	[nan, 'pathog	[0.004384, 0	[0.0, 0.0, 0.0	[0.004384, 0.884357, 0.067013	['benign (high c	-	48497356	48780279	282923	0.98550601
14	TCAF2	chr7	[142415412, 1	['G', 'C',	['T', 'A',	['[2826]', '[2	[1, 1, 1, 1, 1, 1, 1, 1,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.0, 0.0, 0.0	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,	['No prediction f	-	142411223	143836686	1425463	0.9853062
15	GDI2	chr10	[5766802, 577	['C', 'T', '	['A', 'A'	['[2936]', '[2	[1, 1, 1, 1, 1, 1, 10]	[nan,	[0, 0, 0, 0, 0, 0, 0]	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.037726, 0	[0.037726, 0.027517, 0.079609	['benign', 'benig	-	5765222	5813549	48327	0.97755364
16	PTN	chr7	[137227901, 1	['G', 'A',	['T', 'T',	['[2884]', '[2	[1, 5, 7, 1, 1, 1, 1]	[nan,	[0, 0, 0, 0, 0, 0, 0]	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.506611, 0	[0.506611, 0.909667, 0.804208	['pathogenic', 'p	-	137227345	137343733	116388	0.97208002
17	PRKXP1	chr15	[100553584, 1	['C', 'T', '	['G', 'A'	['[2936]', '[2	[1, 5, 1, 1, 6, 1, 1, 1]	[nan,	[0, 1, 0, 0, 0, 0, 0, 0]	[nan, 0	[nan, '	[nar	[nan, 'pathog	[0.0, 0.0, 0.0	[0.771172, 0	[0.771172, 0.533325, 0.017548	['pathogenic', 'p	-	100547751	100559283	11532	0.95900129
18	NEB	chr2	[151492016, 1	['G', 'A',	['T', 'T',	['[2957]', '[2	[1, 1, 1, 1, 1, 1, 1, 4,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.042058, 0	[0.042058, 0.06702, 0.055947,	['benign', 'benig	-	151485338	151734487	249149	0.95818828
19	KIF21A	chr12	[39320015, 39	['T', 'T', '	['A', 'A'	['[2868]', '[2	[1, 7, 1, 1, 7, 1, 1]	[nan,	[0, 0, 0, 0, 0, 0, 0]	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.992293, 0	[0.992293, 0.33121, 0.045525,	['pathogenic (hi	-	39293227	39443120	149893	0.94432766
20	DST	chr6	[56463635, 56	['C', 'G',	['T', 'A',	['[2863, 2863	[1, 1, 1, 1, 1, 1, 10, 1	[nan,	[0, 0, 0, 0, 0, 0, 0, 0	[nan, n	[nan,	[nar	[nan, nan, na	[0.795234, 0	[0.0, 0.0, 0.2	[0.795234, 0.950107, 0.205702	['pathogenic', 'p	-	56457995	56954830	496835	0.94228587
21	DNAH11	chr7	[21561064, 21	['A', 'A',	['T', 'G'	['[2868, 2954	[2, 1, 1, 1, 1, 1, 1, 1,	[nan,	[0, 0, 0, 0, 0, 0, 0, 0, 1	[nan, n	[nan,	[nar	[nan, nan, na	[0.0, 0.0, 0.0	[0.763673, 0	[0.763673, 0.178994, 0.068244	['pathogenic', 'b	+	21543038	21901839	358801	0.92561851

Figure 5.1a: Top twenty genes based on the FATHMM scoring method, Equation 4, in patients with AML.



1	Gene Name	chrom	left	ref_seq	alt_seq	Patient_ID_T	SNV_Tumor_Total	Patient_ID_N	SNV_Normal_Total	Coding Score	Non	Norr	Warr	Coding Score	Non-Coding S	Tumor Total	Warning_y	strand	txStart	txEnd	Length	Gene_Score
2	NOTCH1	chr9	[136496144, 13	['C', 'A',	['A', 'C',	["['PATEAK']	[1, 1, 1, 1, 1, 1, 3, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.878734, 0	[0.0, 0.0, 0.0	[0.878734, 0	['pathogenic	-	136494432	136546048	51616	10.2951979
3	NRAS	chr1	[114709431, 11	['C', 'C',	['A', 'T',	["['PANVGZ'	[1, 1, 1, 3, 1, 2, 1, 6,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.87770	[0.021672, 0	[0.021672, 0	['benign (hig	-	114704468	114716771	12303	6.67373445
4	KRAS	chr12	[25209878, 252	['C', 'C',	['T', 'T', '	["['PARBPX']	[1, 1, 1, 4, 3, 3, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.860194, 0	[0.0, 0.0, 0.0	[0.860194, 0	['pathogenic	-	25205245	25250929	45684	6.56247688
5	CROCCP2	chr1	[16616739, 166	['C', 'C',	['A', 'T',	["['PANUYZ'	[1, 1, 2, 2, 1, 3, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 1, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.0, 0.0	[0.059047, 0	[0.059047, 0	['benign', 'be	-	16618255	16630906	12651	5.1186847
6	FBXW7	chr4	[152322939, 15	['C', 'G',	['T', 'A',	["['PATEAK']	[1, 2, 1, 1, 1, 1, 2, 7,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.887096, 0	[0.0, 0.0, 0.0	[0.887096, 0	['pathogenic	-	152320543	152536092	215549	4.40492889
7	TTN	chr2	[178620341, 17	['C', 'C',	['A', 'A',	["['PARGJY']	[1, 1, 1, 1, 1, 1, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.584471, 0	[0.0, 0.0, 0.0	[0.584471, 0	['pathogenic	-	178525990	178807423	281433	4.03167822
8	BAGE	chr21	[10411922, 104	['G', 'C',	['A', 'T',	["['PATSLH',	[2, 7, 3, 2, 2, 1, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.0, 0.0	[0.063007, 0	[0.063007, 0	['benign', 'be	+	10413519	10454661	41142	3.4059328
9	PCDHGA1	chr5	[141331254, 14	['A', 'C',	['G', 'A',	["['PASFXA']	[1, 1, 1, 1, 1, 1, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.005983, 0	[0.0, 0.0, 0.0	[0.005983, 0	['benign (hig	+	141330684	141512979	182295	2.04847383
10	BAGE2	chr21	[10456288, 104	['C', 'G',	['T', 'T', '	["['PARGMW	[1, 1, 1, 2, 1, 1, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 1, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.0, 0.0	[0.02778, 0.0	[0.02778, 0.0	['benign (hig	+	10413531	10491615	78084	2.00738954
11	TCAF2	chr7	[142407646, 14	['G', 'C',	['T', 'A',	["['PARPIF']"	[1, 1, 1, 1, 1, 2, 2, 1,	[nan, nan, na	[0, 0, 0, 1, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.0, 0.0	[0.0, 0.0, 0.0	[0.0, 0.0, 0.0	['No predicti	+	143620973	143730410	109437	1.94034229
12	PCDHA1	chr5	[140786421, 14	['T', 'C',	['C', 'A',	["['PASUIN']	[1, 1, 1, 1, 1, 1, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.349182, 0	[0.0, 0.0, 0.2	[0.349182, 0	['benign', 'be	+	140786139	141012346	226207	1.93090369
13	TTN-AS1	chr2	[178527821, 17	['C', 'C',	['T', 'A',	["['PASKTG']	[1, 1, 1, 1, 1, 1, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.21248	[0.272967, 0	[0.272967, 0	['benign', 'be	+	178523190	178620217	97027	1.89609756
14	DST	chr6	[56458982, 564	['G', 'G',	['A', 'T',	["['PARBTA']	[1, 1, 1, 1, 1, 1, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 1, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.0, 0.0	[0.619055, 0	[0.619055, 0	['pathogenic	-	56457995	56954830	496835	1.89458744
15	PAX5	chr9	[36840561, 368	['C', 'G',	['A', 'A',	["['PAUBXP']	[1, 1, 1, 1, 1, 1, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.750902, 0	[0.0, 0.04186	[0.750902, 0	['pathogenic	-	36833270	37034268	200998	1.8799582
16	NSD2	chr4	[1918315, 1918	['G', 'C',	['C', 'A',	["['PANTZE']	[1, 1, 1, 1, 1, 1, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 1, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.2575, 0.0,	[0.0, 0.05575	[0.2575, 0.05	['benign', 'be	+	1871392	1982192	110800	1.86370302
17	UBR4	chr1	[19078157, 190	['C', 'C',	['A', 'T',	["['PARMWZ	[1, 1, 1, 1, 1, 1, 1, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.0, 0.0	[0.059293, 0	[0.059293, 0	['benign', 'be	-	19074510	19210266	135756	1.79029269
18	FLT3	chr13	[28014333, 280	['C', 'C',	['A', 'A',	["['PAPAGW	[1, 1, 1, 1, 1, 1, 2, 1,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.0, 0.5	[0.016535, 0	[0.016535, 0	['benign (hig	-	28003273	28100576	97303	1.75805114
19	MUC4	chr3	[195747038, 19	['C', 'T',	['T', 'C', '	["['PANXCX']	[1, 1, 1, 1, 3, 2, 1, 1,	[nan, nan, "[	[0, 0, 1, 0, 0, 0, 0, 0,	[nan, nan, 0.	[nar	[nan	[nan,	[0.0, 0.0, 0.0	[0.126804, 0	[0.126804, 0	['benign', 'be	-	195746765	195811973	65208	1.73569015
20	PTPN11	chr12	[112446093, 11	['C', 'G',	['A', 'A',	["['PANSHK']	[1, 1, 1, 1, 1, 1, 2, 2,	[nan, nan, na	[0, 0, 0, 0, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.91094	[0.042105, 0	[0.042105, 0	['benign', 'pa	+	112418946	112509918	90972	1.67980927
21	MIR1244-1	chr12	[9239313, 9239	['T', 'G',	['C', 'T', '	["['PARMKK'	[1, 1, 3, 2, 4, 4, 1, 1,	[nan, nan, na	[0, 0, 0, 1, 0, 0, 0, 0,	[nan, nan, na	[nar	[nan	[nan,	[0.0, 0.0, 0.0	[0.050976, 0	[0.050976, 0	['benign', 'pa	-	9239469	9239551	82	1.67187642

Figure 5.2: Top twenty genes based on the FATHMM scoring method, Equation 4, in patients with ALL.

From these figures, the top twenty genes can be cross-referenced to the known AML- and ALL-associated genes listed in Appendix A.1. None of the 20 high-scoring AML genes appear within the list in Appendix A.1. However, the ALL-highest score gene list has 6 hits that correspond with the list in Appendix A.2. The lack of overlap between the identified high-scoring AML genes with the known genes associated with the disease suggest that our scoring scheme can be improved. We anticipate that incorporating PROVEAN, STRUM, and SNPnexus assessments of individual SNVs will bring about a more a balanced evaluation of the genes and lead to better predictions.

Step 3. Bioinformatics analyses: GO terms, pathways, and protein-protein interactions

Looking into the high scoring genes found in Step 2, we can conduct a more detailed bioinformatics investigation on them to find what biological processes they are involved in. This can be done via the following analysis:

(i) Finding enriched GO terms

Gene Ontology is a web-based server in which its purpose is to provide gathered knowledge about the functions of genes. This site contains biological information from the molecular level to larger pathway on cellular and organism level systems. With the evolutionary idea, that inherited genes across organisms can share homologous genes due to a common ancestor, the research performed on the biological process of a shared gene in one organism can be applicable to others. When investigating large scale data, on feature provided is the ability to cluster different kinds of biological functions into groups but can also indicate how these groups relate to each other [47].

(ii) Identifying molecular pathways

To do so, Kyoto Encyclopedia of Genes and Genomes (KEGG), a databased resource can be used. The key feature that will be used on this resource is the KEGG Pathway Maps section, by using an interaction and reaction network diagrams, it generalizes genomic information among organisms. With their in-house software, KegSketch, a map can be manually drawn and delivers an output KGML+ file is created containing graphics of the KEGG objects that can be mapped using its object identifiers resource link [48].

(iii) Constructing protein-protein interaction networks.

STRING [49] allows the user to input a list of genes, for this study it will be the high scoring genes, and the output is an analysis of how these nodes interact, or not, to each other. It creates a clustering mechanism that groups together the genes that have a connection and isolates that genes that have no correlation to each other. Ideally the clustered genes are the ones that will be used for the interpretation, whereas the stand-alone genes are typically disregarded. Additional information that STRING provides in their output is biological process, links to publication where they have been mentioned, KEGG pathways, and their local network. With the listed publications of articles these genes have been associated with, it will assist with the discovery of genes that have no known connection to patients with leukemia.

Throughout the gene selection process and the downstream bioinformatics analyses, particular attention will be paid to the kinase, phosphatase, and glycosyltransferase genes because there are interests from collaborators who can conduct wet-lab experiments to investigate their biological roles in cancer. Indeed, some studies have shown the effect variants altering kinases and phosphatases links directly to cancer. Being that kinases and phosphatases are responsible for the post-translational modification of proteins, mutations in these can disrupt the cellular signaling pathway. The imbalance between kinases and phosphatases can also suppress the response to cancer treatments and decrease the likelihood of survival [24]. Glycosyltransferase, also involved in post- translational modification, prepare and transfer glycan chains on mucins to the correct destination [25]. Abnormalities in these genes can alter their expression, thus being a driving force for cancer development and progression [46].

I will start with compiling the list of human protein kinase, phosphatase, and glycosyltransferase genes. Once this list is compiled, a Python code can be used to identify the genes that were found in this study as one of the three human proteins that are being focused on. While all three are involved in post-translational modification, after the identification process, these genes can be then properly grouped and their role in AML and ALL can be assessed.

5.2.2 Relating survival outcomes with demographic, clinical, and genetic variables

For this specific aim, we want to identify a predictive model relating the patients' overall survival time to their demographic, clinical, and genomic mutation information using suitable regression type analysis or other statistical and machine learning techniques. This model development will be done in collaboration with a Data Science PhD student in our group, Mr. Kelvin Ofori-Minta, who is developing the model for a similar set of data for patients with prostate cancer.

After selecting the high-scoring genes for AML and ALL from specific aim 1, we will add them to the lists of known AML- and ALL-associated genes in Appendix A. The new list of AML and ALL genes, expected to contain around 100 in total, will be added as variables (columns) to the respective data files that contain the patients' demographic and clinical information. Under the column for each gene variable, we will enter at each row (patient) the count of SNVs found within that particular gene in the patient.

The overall survival time, with survival status (alive or deceased) as censoring indicator, will be considered the response variable. All other variables in the integrated demographic-clinical-genetic dataset will form the initial set of predictor variables for the regression model. We will manually examine the column headings of the demographic and clinical variables to remove any uninformative variables such as those with too many missing data values or those with no variation from patient to patient. With the remaining variables, a simple pairwise correlation analysis will be conducted. If a group of variables are found to be highly correlated, only one of them will be selected as representative to be included in the model and others discarded.

The computational methods developed for the various analyses in my research project will be implemented in R and Python. These newly developed programs will be added to our existing OncoMiner Pipeline as new modules so that they can be publicly accessed and utilized by other researchers.

5.3 Timeline

Table 5.1: 2 Year Timeline	
Summer 2023	• Thesis/PhD Proposal defense • Attend and present at the ISCB 2023 international conference • Identify the kinase, phosphatase, and glycosyltransferase genes
Fall 2023	• Explore the use of STRUM and SNPnexus in scoring functional effects of individual SNVs • Incorporate structural changes predicted by STRUM to develop a new scoring function for genes • Attend and present at SC23 conference and cancer workshop
Spring 2024	• Implement new gene scoring function as a module to add on to the existing Oncominer Pipeline • Beta testing of new module Apply new module to identify novel ALL- and AML-associated genes
Summer 2024	• Finalize new OncoMiner module go live on UTEP website • Prepare and submit manuscript on new scoring function for publication • Begin analysis of survival outcomes in relation to clinical, demographic, and genetic variables.
Fall 2024	• GO term, pathway, protein interaction analyses for identified ALL- and AML-associated genes • Finish modeling for survival outcomes with assessment of predictive power. • First complete dissertation draft ready • Committee meeting to go over main results
Spring 2025	• Prepare and submit manuscript on findings of the above bioinformatics analyses • Revise dissertation and prepare final draft for defense

References

[1] Acute *myeloid Leukemia (AML) subtypes and prognostic factors*. American Cancer Society. (2018, August 21). Retrieved January 13, 2023, from https://www.cancer.org/cancer/acute-myeloid-leukemia/detection-diagnosis-staging/how-classified.html

[2] “Leukemia - Cancer Stat Facts.” *SEER*, https://seer.cancer.gov/statfacts/html/leuks.html.

[3] Vakiti, Anusha, and Prerna Mewawalla. “NCBI Bookshelf.” *Acute Myeloid Leukemia*, StatPearls Publishing LLC, 2022, https://www.ncbi.nlm.nih.gov/books/NBK507875/.

[4] Kumar, C Chandra. “Genetic Abnormalities and Challenges in the Treatment of Acute Myeloid Leukemia.” *Genes & Cancer*, U.S. National Library of Medicine, Feb. 2011, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3111245/.

[5] “Lymphocyte.” *Genome.gov*, 26 Jan. 2023, https://www.genome.gov/genetics-glossary/Lymphocyte.

[6] Karen Seiter, MD. “Acute Lymphoblastic Leukemia (ALL) Staging.” *Classification for Acute Lymphoblastic Leukemia*, Medscape, 1 Nov. 2022, https://emedicine.medscape.com/article/2006661-overview.

[7] *Acute Lymphoblastic Leukemia - My Cancer Genome*, https://www.mycancergenome.org/content/disease/acute-lymphoblastic-leukemia/.

[8] Yu, Jifeng, et al. “Advances in Targeted Therapy for Acute Myeloid Leukemia - Biomarker Research.” *BioMed Central*, BioMed Central, 20 May 2020, https://biomarkerres.biomedcentral.com/articles/10.1186/s40364-020-00196-2.

[9] “Acute Myeloid Leukemia.” *Acute Myeloid Leukemia - My Cancer Genome*, https://www.mycancergenome.org/content/disease/acute-myeloid-leukemia/.

[10] Jeong, Ji Hun, et al. “A Case of Therapy-Related Acute Myeloid Leukemia with Inv(16)(p13.1q22) after Single Low-Dose Iodine-131 Treatment for Thyroid Cancer.” *The Korean Journal of Hematology*, vol. 47, no. 3, 2012, p. 225., https://doi.org/10.5045/kjh.2012.47.3.225.

[11] Rogers, Mark F, et al. “Fathmm-XF: Accurate Prediction of Pathogenic Point Mutations via Extended Features.” *Bioinformatics (Oxford, England)*, U.S. National Library of Medicine, 1 Feb. 2018, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5860356/.

[12] Chauhan, D. T. (2021, September 13). *What is the importance of GC content?* Genetic Education. Retrieved March 1, 2023, from https://geneticeducation.co.in/what-is-the-importance-of-gc-content/

[13] *The causes of mutations - understanding evolution*. Understanding Evolution - Your one-stop source for information on evolution. (2022, September 9). https://evolution.berkeley.edu/evolution-101/mechanisms-the-processes-of-evolution/the-causes-of-mutations/

[14] Ray, P. D., Huang, B.-W., & Tsuji, Y. (2012, May). *Reactive oxygen species (ROS) homeostasis and redox regulation in cellular signaling*. Cellular signalling. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3454471/#R1

[15] *Cancer staging: Has cancer spread: Cancer prognosis*. Has Cancer Spread | Cancer Prognosis. (n.d.). https://www.cancer.org/cancer/diagnosis-staging/staging.html

[16] Lagunas-Rangel, F. A., Chávez-Valencia, V., Gómez-Guijosa, M. Á., & Cortes-Penagos, C. (2017, October 1). Acute myeloid leukemia-genetic alterations and their clinical prognosis. International journal of hematology-oncology and stem cell research. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5767295/#:~:text=The%20fusion%20gene%20MLLT3%2DMLL,through%20the%20remodeling%20of%20chromatin.

[17] Molecular genetic markers in acute myeloid leukemia - PMC. (n.d.). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4470139/

[18] Gao, J., Gentzler, R. D., Timms, A. E., Horwitz, M. S., Frankfurt, O., Altman, J. K., & Peterson, L. C. (2014, April 22). Heritable GATA2 mutations associated with familial AML-MDS: A case report and review of literature - journal of hematology & oncology. SpringerLink. https://link.springer.com/article/10.1186/1756-8722-7-36

[19] Yagi, T., Morimoto, A., Eguchi, M., Hibi, S., Sako, M., Ishii, E., Mizutani, S., Imashuku, S., Ohki, M., & Ichikawa, H. (2003, September 1). Identification of a gene expression signature associated with pediatric AML Prognosis. American Society of Hematology. https://ashpublications.org/blood/article/102/5/1849/17538/Identification-of-a-gene-expression-signature

[20] Entry - #601626 - leukemia, acute myeloid; AML - OMIM. (n.d.-a). https://www.omim.org/entry/601626

[21] Iacobucci, I., & Mullighan, C. G. (2017, March 20). Genetic basis of acute lymphoblastic leukemia. Journal of clinical oncology : official journal of the American Society of Clinical Oncology. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5455679/

[22] Entry - %613065 - leukemia, acute lymphoblastic; all - OMIM. (n.d.-a). https://www.omim.org/entry/613065

[23] Wang, B. (n.d.). *Bofei-Wang/ProstateCancer_TCGA_VCF*. GitHub. https://github.com/bofei-wang/ProstateCancer_TCGA_VCF

[24] Turdo, A., D'Accardo, C., Glaviano, A., Porcelli, G., Colarossi, C., Colarossi, L., Mare, M., Faldetta, N., Modica, C., Pistone, G., Bongiorno, M. R., Todaro, M., & Stassi, G. (2021, October 4). *Targeting phosphatases and kinases: How to checkmate cancer*. Frontiers. https://www.frontiersin.org/articles/10.3389/fcell.2021.690306/full

[25] Amado, M., Argüeso, P., Bobek, L. A., Breton, C., Chao, C. C., Corfield, A. P., Cotsarelis, G., DeSouza, M. M., Gipson, I. K., Godl, K., Gum, J. R., Gururaja, T. L., Hanna, C., Ho, S. B., Holly, F. J., Inatomi, T., Jono, H., Komatsu, M., Lan, M. S., … Dartt, D. A. (2004, January 7). *Role of mucins in the function of the corneal and conjunctival epithelia*. International Review of Cytology. https://www.sciencedirect.com/science/article/abs/pii/S0074769603310010

[26] Deak, D., Gorcea-Andronic, N., Sas, V., Teodorescu, P., Constantinescu, C., Iluta, S., Pasca, S., Hotea, I., Turcas, C., Moisoiu, V., Zimta, A.-A., Galdean, S., Steinheber, J., Rus, I., Rauch, S., Richlitzki, C., Munteanu, R., Jurj, A., Petrushev, B., … Tomuleasa, C. (2021, January). *A narrative review of central nervous system involvement in acute leukemias*. Annals of translational medicine. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7859772/#:~:text=In%20acute%20lymphoblastic%20leukemia%20(ALL,leukemia%20(15%2C16).

[27] Lee, S. (n.d.). *Phases of childhood leukemia*. Canadian Cancer Society. https://cancer.ca/en/cancer-information/cancer-types/leukemia-childhood/staging

[28] Thastrup, M., Duguid, A., Mirian, C., Schmiegelow, K., & Halsey, C. (2022, October 20). *Central Nervous System involvement in childhood acute lymphoblastic leukemia: Challenges and solutions*. Nature News. https://www.nature.com/articles/s41375-022-01714-x

[29] *What is next-generation sequencing?: Thermo Fisher Scientific - US*. What is Next-Generation Sequencing? | Thermo Fisher Scientific - US. (n.d.). https://www.thermofisher.com/us/en/home/life-science/sequencing/sequencing-learning-center/next-generation-sequencing-information/ngs-basics/what-is-next-generation-sequencing.html

[30] *AACR Project GENIE®: Powering Precision Medicine*. American Association for Cancer Research (AACR). (2023, April 5). https://www.aacr.org/professionals/research/aacr-project-genie/

[31] AACRGENIE13.0-publicDataGuide - American Association for Cancer Research. (n.d.). https://www.aacr.org/wp-content/uploads/2023/03/13.0_data_guide-1.pdf

[32] *Therapeutically applicable research to generate effective treatments (target)*. ccg - National Cancer Institute. (n.d.). https://www.cancer.gov/ccg/research/genome-sequencing/target

[33] *The cancer genome atlas program (TCGA)*. ccg - National Cancer Institute. (n.d.-a). https://www.cancer.gov/ccg/research/genome-sequencing/tcga

[34] *The Next Generation Cancer Knowledge Base*. Home | NCI Genomic Data Commons. (n.d.). https://gdc.cancer.gov/#:~:text=The%20GDC%20provides%20tools%20to,large%2C%20high%20volume%20molecular%20data.

[35] Welcome to the documentation for cBioPortal! (n.d.). https://docs.cbioportal.org/

[36] Chromosome lengths. (n.d.). http://www.insilicase.com/Web/Chromlen.aspx

[37] *UCSC Genome Annotation*. Index of /goldenpath/hg38/database. (2013, December). https://hgdownload.cse.ucsc.edu/goldenpath/hg38/database/

[38] Rogers MF, Shihab HA, Mort M, Cooper DN, Gaunt TR, Campbell C. FATHMM-XF: enhanced accuracy in the prediction of pathogenic sequence variants via an extended feature set, Bioinformatics, September 2017.

[39] *Provean*. J. Craig Venter Institute. (n.d.). https://www.jcvi.org/research/provean#downloads

[40] Jorge Oscanoa, Lavanya Sivapalan, Emanuela Gadaleta, Abu Z Dayem Ullah, Nicholas R Lemoine, Claude Chelala, SNPnexus: a web server for functional annotation of human genome sequence variation (2020 update), Nucleic Acids Research, 2020, 48(W1):W185-W192.

[41] Abu Z Dayem Ullah, Jorge Oscanoa, Jun Wang, Ai Nagano, Nicholas Lemoine, Claude Chelala, SNPnexus: assessing the functional relevance of genetic variation to facilitate the promise of precision medicine, Nucleic Acids Research, 2018, 46(W1):W109-W113.

[42] Abu Z Dayem Ullah, Nicholas R Lemoine and Claude Chelala, A practical guide for the functional annotation of genetic variations using SNPnexus, Briefings in Bioinformatics, 2013, 14(4):437-47.

[43] Abu Z Dayem Ullah, Nicholas R Lemoine and Claude Chelala, SNPnexus: a web server for functional annotation of novel and publicly known genetic variants (2012 update), Nucleic Acids Research, 2012, 40(W1):W65-W70.

[44] Claude Chelala, Arshad Khan and Nicholas R Lemoine, SNPnexus: A web database for functional annotation of newly discovered and public domain Single Nucleotide Polymorphisms, Bioinformatics, 2009, 25(5):655-61.

[45] University of Michigan. (n.d.). *Strum: Structure-based stability change prediction upon single-point mutation*. Zhang Lab. https://zhanggroup.org/STRUM/

[46] Bhullar, K. S., Lagar&oacute;n, N. O., McGowan, E. M., Parmar, I., Jha, A., Hubbard, B. P., & Rupasinghe, H. P. V. (2018, February 19). *Kinase-targeted cancer therapies: Progress, challenges and future directions - molecular cancer*. BioMed Central. https://molecular-cancer.biomedcentral.com/articles/10.1186/s12943-018-0804-2

[47] *About the GO*. Gene Ontology Resource. (n.d.). http://geneontology.org/docs/introduction-to-go

[48] Kegg Pathway Maps. (n.d.). https://www.genome.jp/kegg/kegg3a.html

[49] STRING consortium. (n.d.). *Welcome to string*. STRING. https://string-db.org/cgi/input?sessionId=bka63caCd96D&input_page_active_form=multiple_identifiers

[50] Wang, Bofei, "Identification Of Prostate Cancer-Associated Genomic Alterations By Analyzing Variant Frequencies, Functional Effects, And Protein Interactions" (2021). *Open Access Theses & Dissertations*. 3459. https://scholarworks.utep.edu/open_etd/3459

[51] Singh, S. K., Lupo, P. J., Scheurer, M. E., Saxena, A., Kennedy, A. E., Ibrahimou, B., Barbieri, M. A., Mills, K. I., McCauley, J. L., Okcu, M. F., & Dorak, M. T. (2016). A childhood acute lymphoblastic leukemia genome-wide association study identifies novel sex-specific risk variants. *Medicine, 95*(46). https://doi.org/10.1097/md.0000000000005300

APPENDIX A

A.1 List of acute myeloid leukemia associated genes found in literature

Gene Name	Reference Number
PML	[16]
RARA	[16]
DNMT1	[16]
DNMT3	[16]
MLL	[16]
MLLT3	[16]
H3K4	[16]
DOT1L	[16]
MENIN1	[16]
HOX	[16]
DEK	[16]
NUP214	[16]
FLT3	[16]
EVI1	[16]
HSC	[16]
DATA2	[16]
PBX1	[16]
PLM	[16]
RPN1	[16]
RBM15	[16]
MKL1	[16]
KIT	[16]
NPM1	[16]
CEBPA	[16]
RAS	[16]
WT1	[16]
BAALC	[16]
ERG	[16]
MN1	[16]
TET2	[16]
IDH	[16]
ASXL1	[16]
PTPN11	[16]
CBL	[16]

ARF	[16]
RUNX1	[17]
CBFB	[17]
MYH11	[17]
RMB15	[17]
BCOR	[17]
KMT2A	[17]
NRAS	[17]
KRAS	[17]
PHF6	[17]
TP53	[17]
GATA2	[18]
VDAC1	[19]
ZYX	[19]
VAT1	[19]
NPC2	[19]
AZU1	[19]
HOMER-3	[19]
PGD	[19]
ENSA	[19]
TKT	[19]
BST1	[19]
STK17B	[19]
CDK6	[19]
RAB32	[19]
PTP4A2	[19]
APLP2	[19]
CYLN2	[19]
OGT	[19]
HNRPD	[19]
POLR2H	[19]
TIAL1	[19]
ATP6F	[19]
NME1	[19]
GRIK5	[19]
CD14	[19]
GABARAP	[19]
NFKBIA	[19]
GCN5L2	[19]

HSPE1	[19]
LBR	[19]
ECE2	[19]
BZRP	[19]
XPO1	[19]
HIPK2	[20]
SF3B1	[20]
TERC	[20]
TERT	[20]
FPDMM	[20]
PFBMFT1	[20]
PFBMFT2	[20]
SMMHC	[20]
AF9	[20]
BRD4	[20]
WNT	[20]
BET	[20]
CXXC6	[20]
MCL1	[20]
DNMT3A	[20]

A.2 List of acute lymphoblastic leukemia associate genes found in literature

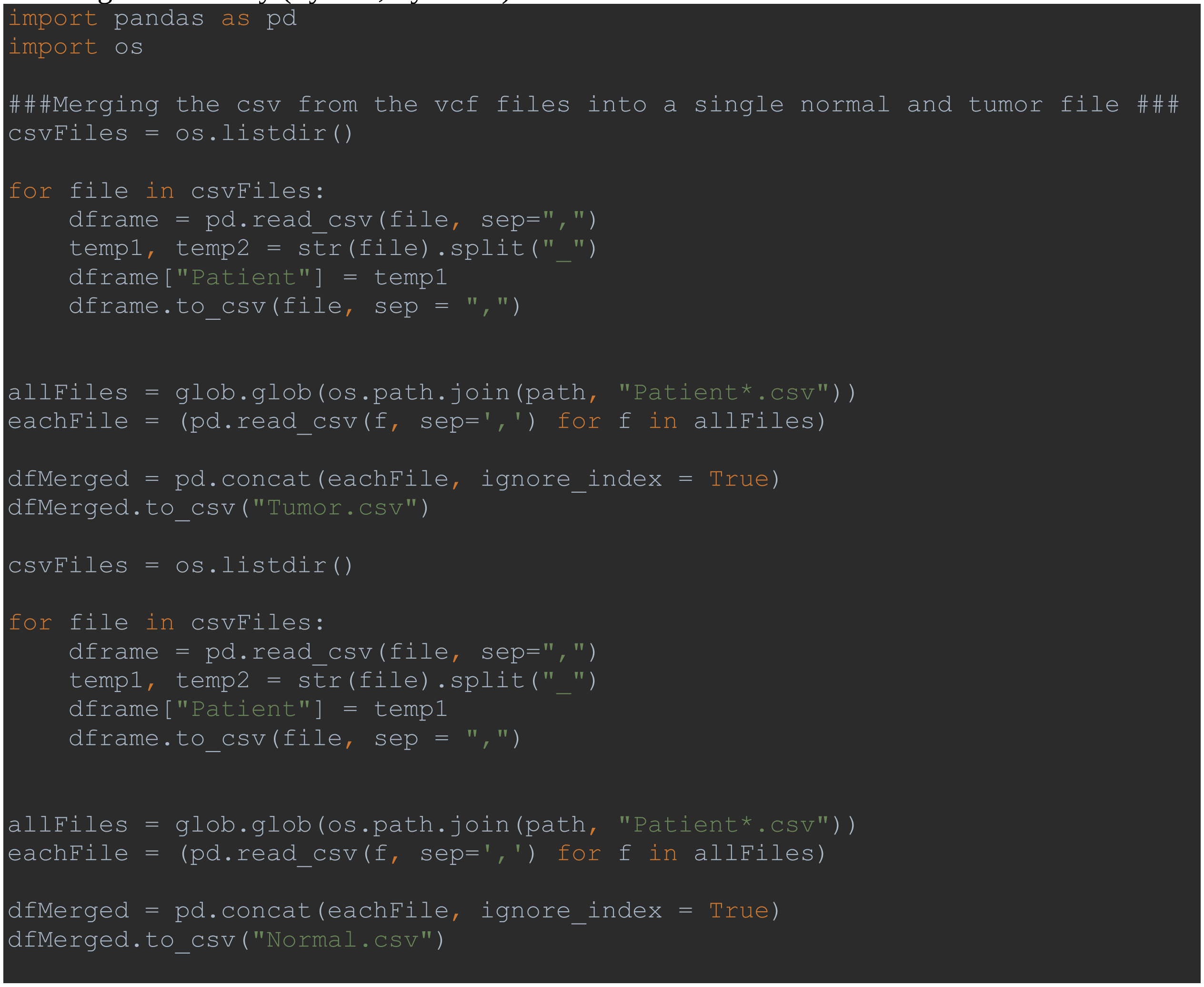
Gene Name	**Reference Number**
RAS	[21]
IKZF3	[21]
TP53	[21]
CDKN2A	[21]
CDKN2B	[21]
RB1	[21]
IKZF2	[21]
ETV6	[21]
RUNX1	[21]
TCF3	[21]
PBX1	[21]
IKZF1	[21]
PAX5	[21]
CRLF2	[21]

JAK1	[21]
JAK2	[21]
PI3K	[21]
P2RY8	[21]
KMT2A	[21]
MLL	[21]
DUX4	[21]
MEF2D	[21]
ZNF384	[21]
TAL1	[21]
LMO2	[21]
TLX1	[21]
TLX2	[21]
NUP214	[21]
ABL1	[21]
NOTCH1	[21]
FBXW7	[21]
EML1	[21]
HOX11	[21]
HOX11L2	[21]
HLF	[21]
EBF1	[21]
ERG	[21]
PTPN11	[21]
NF1	[21]
FLT3	[21]
NTRK3	[21]
BLNK	[21]
TYK2	[21]
PTK2B	[21]
EPOR	[21]
EP300	[21]
HDAC9	[21]
ID4	[21]
ARID1B	[21]
SYNRG	[21]
EWSR1	[21]
CREBBP	[21]
TAF15	[21]

BCR-ABL1	[21]
RAG	[22]
MGA	[22]
ATF7IP	[22]
TXNIP	[22]
CNR2	[22]
AMPK	[22]
ITGA6	[22]
BAX	[22]
LEF1	[22]
IL7R	[22]
TLX3	[22]
PHF6	[22]
NCOR1	[22]
SPI1	[22]
TCF4	[22]
TCF7L2	[22]
CTCF	[22]
STAT5	[22]
HOXA	[22]
SUZ12	[22]
EZH2	[22]
PRC2	[22]
CNOT3	[22]
RPL5	[22]
RPL10	[22]
NSD2	[22]
GNB1	[22]
NT5C2	[22]
DNM2	[22]
RELN	[22]
ECT2L	[22]
EED	[22]
SETD2	[22]
GATA3	[22]
SH2B3	[22]

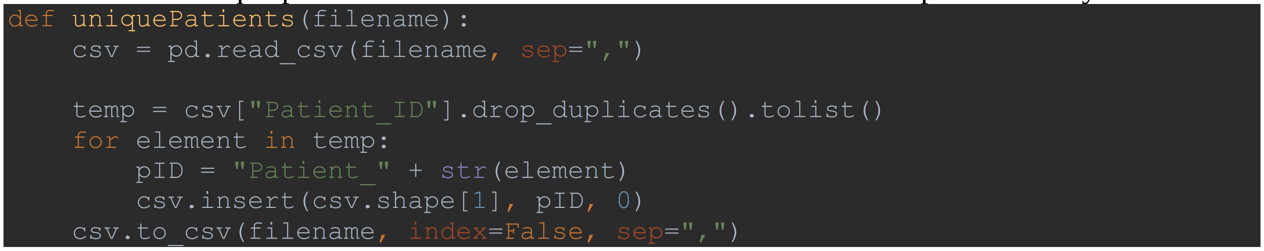
APPENDIX B

B.1 Post VCF to CSV conversion, this script merges all the normal and tumor CSV files within the assigned directory (Python, Pycharm).



```
import pandas as pd
import os

###Merging the csv from the vcf files into a single normal and tumor file ###
csvFiles = os.listdir()

for file in csvFiles:
    dframe = pd.read_csv(file, sep=",")
    temp1, temp2 = str(file).split("_")
    dframe["Patient"] = temp1
    dframe.to_csv(file, sep = ",")


allFiles = glob.glob(os.path.join(path, "Patient*.csv"))
eachFile = (pd.read_csv(f, sep=',') for f in allFiles)

dfMerged = pd.concat(eachFile, ignore_index = True)
dfMerged.to_csv("Tumor.csv")

csvFiles = os.listdir()

for file in csvFiles:
    dframe = pd.read_csv(file, sep=",")
    temp1, temp2 = str(file).split("_")
    dframe["Patient"] = temp1
    dframe.to_csv(file, sep = ",")


allFiles = glob.glob(os.path.join(path, "Patient*.csv"))
eachFile = (pd.read_csv(f, sep=',') for f in allFiles)

dfMerged = pd.concat(eachFile, ignore_index = True)
dfMerged.to_csv("Normal.csv")
```

B.2 Finds the unique patients in the dataset and inserts columns for the patient binary matrix.



```
def uniquePatients(filename):
    csv = pd.read_csv(filename, sep=",")

    temp = csv["Patient_ID"].drop_duplicates().tolist()
    for element in temp:
        pID = "Patient_" + str(element)
        csv.insert(csv.shape[1], pID, 0)
    csv.to_csv(filename, index=False, sep=",")
```

B.3 Changes the entries of the patient binary matrix to 1 if the variant is present in patient

```
def variant(filename):
    csv = pd.read_csv(filename, sep=",")
    csv["Patient_ID"] = csv["Patient_ID"].to_list()
    for row in range(csv.shape[0]):
        temp = csv["Patient_ID"][row].replace("[", "")
        temp = temp.replace("'", "")
        temp = temp.replace("]", "")
        temp = temp.replace(" ", "")
        temp = temp.split(",")
        for element in temp:
            # print(element)
            name = "Patient_" + element
            csv.loc[row, name] = 1
    newName = "Final_" + filename
    csv.to_csv(newName, sep=",")
```

B.4 Combines the normal and tumor csv files into a single csv with the binary matrix.

```
####### Combine normal and tumor binary matrices for AML ###############
columns = list(amlNorm.columns)

for element in columns:
    temp = "Normal_" + element
    amlNorm = amlNorm.rename(columns={element : temp})


aml = amlTumor.merge(amlNorm, left_on=["chrom", "left", "ref_seq", "alt_seq",
"New_Gene_Name"], right_on=["Normal_chrom", "Normal_left", "Normal_ref_seq",
"Normal_alt_seq", "Normal_New_Gene_Name"], how="outer")

aml.to_csv("Final_AML.csv", index=False)


# Do the same for ALL

allNorm = pd.read_csv("Final_New_ALL_Norma_Prep.csv", sep=",")
allTumor = pd.read_csv("Final_New_ALL_Tumor_Prep.csv", sep=",")

columns = list(allNorm.columns)

for element in columns:
    temp = "Normal_" + element
    allNorm = allNorm.rename(columns={element : temp})


all = allTumor.merge(allNorm, left_on=["chrom", "left", "ref_seq", "alt_seq",
"New_Gene_Name"], right_on=["Normal_chrom", "Normal_left", "Normal_ref_seq",
"Normal_alt_seq", "Normal_New_Gene_Name"], how = "outer")

all.to_csv("Final_ALL.csv", index=False)
```

B.5 Returns the common variants found between the normal and tumor, prints this value.

```
def common(file1, file2):
    csv1 = pd.read_csv(file1, sep=",")
    csv2 = pd.read_csv(file2, sep=",")

    # Concatenate the two csv get a total count of both normal and tumor
together
    csv = pd.concat([csv1, csv2])
    all = len(csv)
    # drop only the common rows from the total to get the new shape the whole
without the overlap
    newcsv = csv.drop_duplicates(subset=["chrom", "left", "ref_seq",
"alt_seq", "New_Gene_Name"], keep=False)
    uncom = len(newcsv)
    return print(all, uncom)
```

B.6 Creation of the translation dictionary for genes found on the negative strand of DNA (Python, Pycharm).

```
conversion_dictionary = {
    "A": "T",
    "T": "A",
    "U": "A",
    "G": "C",
    "C": "G",
    "Y": "R",
    "R": "Y",
    "N": "N"
}
```

B.7 Function that takes a variant with many isoforms and in a new column assigns the first isoform as this value (Python, Pycharm)..

```
def condense(filename, column):
    csv = pd.read_csv(filename)
    newCol = []
    for row in range(csv.shape[0]):
        temp = csv[column][row].replace("[", "")
        temp = temp.replace("'", "")
        temp = temp.replace("]", "")
        temp = temp.replace(" ", "")
        temp = temp.split(",")
        newCol.append(temp[0])
    newName = "New_" + column
    csv.insert(csv.shape[1], newName, newCol)
    csv.to_csv(filename, sep=",", index=False)
```

B.8 Function that writes in text file that exact format required for the FATHMM input (Python, Pycharm).

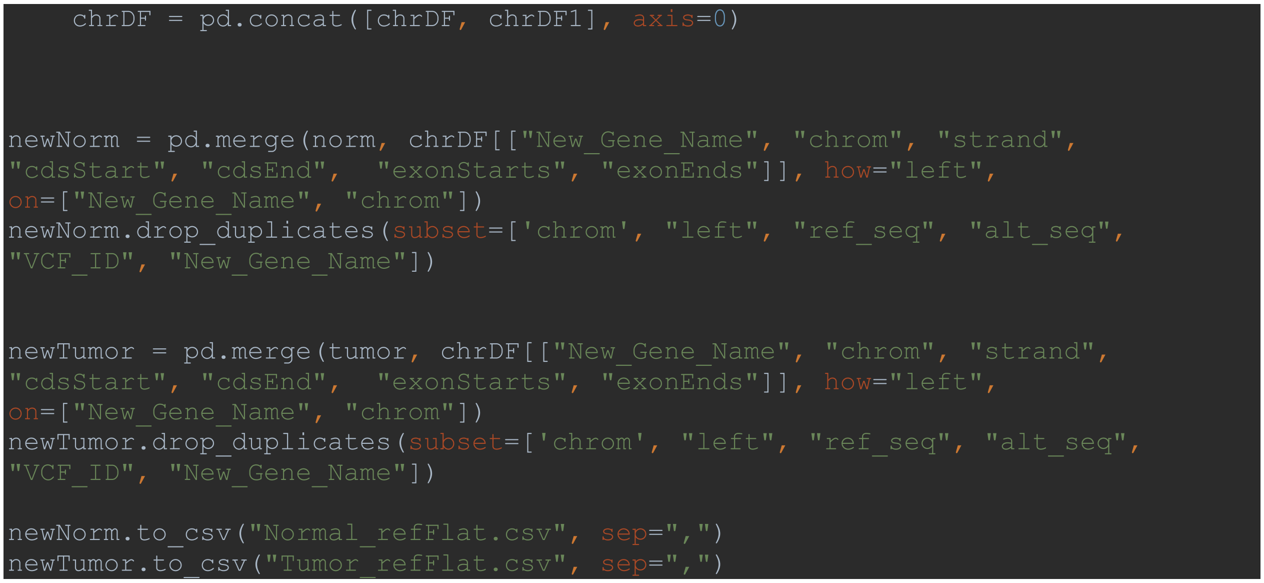
```
def fathmmFormat(csv, txtfile):
    with open(txtfile, 'w') as f:
        for row in range(csv.shape[0]):
            chrom = csv.loc[row, "chrom"][3:]
            left = str(csv.loc[row, "left"])
            ref = csv.loc[row, "ref_seq"]
            alt = csv.loc[row, "alt_seq"]
            temp = chrom + "," + left + "," + ref + "," + alt
            f.write(temp)
            f.write("\n")
    f.close()
```

B.9 Isolates only the gene information and removes duplicates, resulting in a unique gene list for both normal and tumor (Python, Pycharm)

```
############ Isolating the Gene info only ############
norm = pd.read_csv("Normal_refFlat.csv")
tumor = pd.read_csv("Tumor_refFlat.csv")

geneNorm = norm[["New_Gene_Name", "chrom", "strand", "cdsStart", "cdsEnd",
"exonStarts", "exonEnds"]]
geneTumor = tumor[["New_Gene_Name", "chrom", "strand", "cdsStart", "cdsEnd",
"exonStarts", "exonEnds"]]

geneNorm.to_csv("Normal_Gene.csv", sep=",")
geneTumor.to_csv("Tumor_Gene.csv", sep=",")

####################### Removing duplicated genes ###########################
norm = norm.drop_duplicates()
tumor = tumor.drop_duplicates()

norm.to_csv("Normal_Gene.csv", sep=",")
tumor.to_csv("Tumor_Gene.csv", sep=",")
```

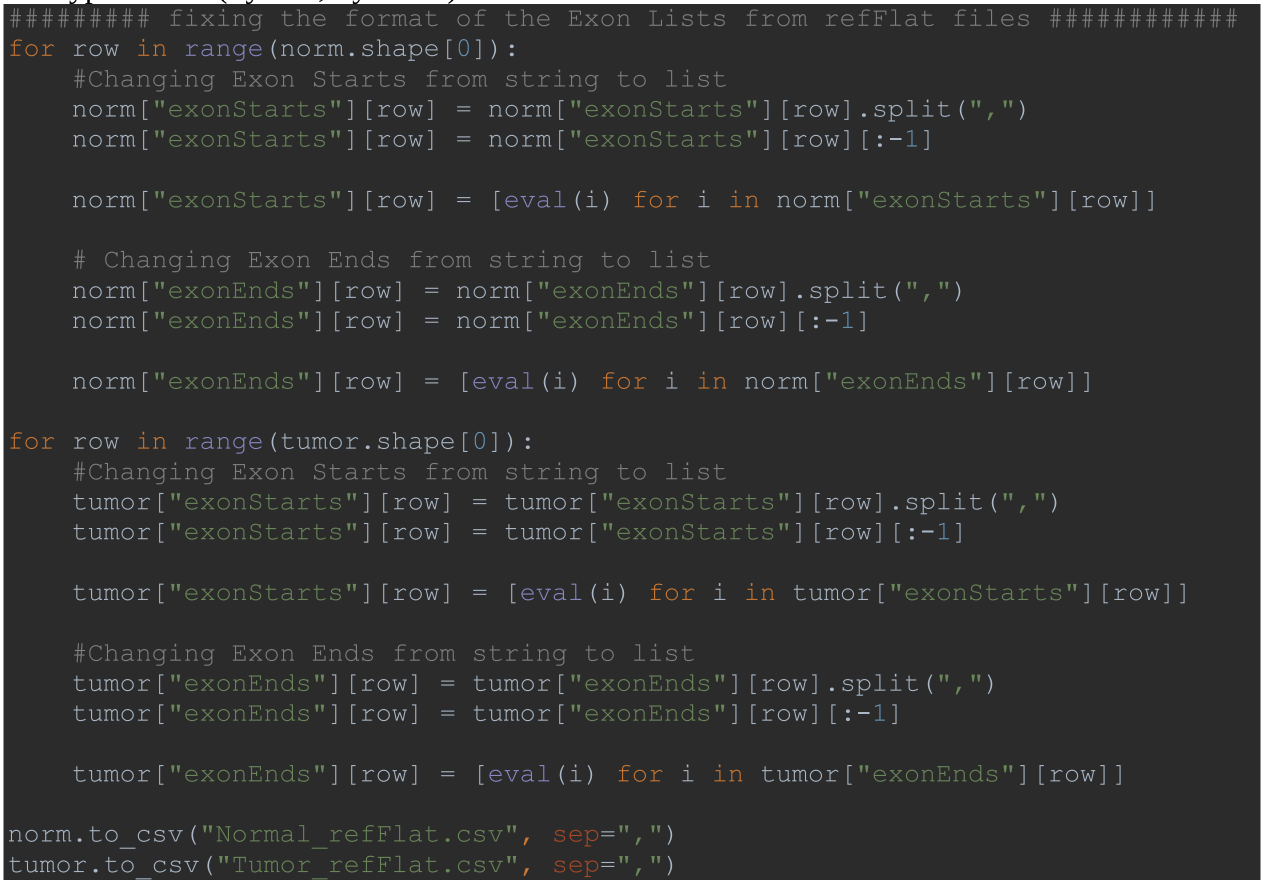
B.10 Creates a dataframe from all the refFlat files, pointing to the directory where they are saved, and merges specified columns to the unique gene lists

```
#### Adding specified information from the refFlat files onto the OMI ######
refFlatfiles = os.listdir()

chrDF = pd.DataFrame(columns=["New_Gene_Name", "name", "chrom", "strand",
"txStart", "txEnd", "cdsStart", "cdsEnd", "exonCount", "exonStarts",
"exonEnds"])
for file in refFlatfiles:

    chrDF1 = pd.read_csv(file, sep="\t", header=None, names=["New_Gene_Name",
"name", "chrom", "strand", "txStart", "txEnd", "cdsStart", "cdsEnd",
"exonCount", "exonStarts", "exonEnds"])
```



```
    chrDF = pd.concat([chrDF, chrDF1], axis=0)



newNorm = pd.merge(norm, chrDF[["New_Gene_Name", "chrom", "strand",
"cdsStart", "cdsEnd",  "exonStarts", "exonEnds"]], how="left",
on=["New_Gene_Name", "chrom"])
newNorm.drop_duplicates(subset=['chrom', "left", "ref_seq", "alt_seq",
"VCF_ID", "New_Gene_Name"])


newTumor = pd.merge(tumor, chrDF[["New_Gene_Name", "chrom", "strand",
"cdsStart", "cdsEnd",  "exonStarts", "exonEnds"]], how="left",
on=["New_Gene_Name", "chrom"])
newTumor.drop_duplicates(subset=['chrom', "left", "ref_seq", "alt_seq",
"VCF_ID", "New_Gene_Name"])

newNorm.to_csv("Normal_refFlat.csv", sep=",")
newTumor.to_csv("Tumor_refFlat.csv", sep=",")
```

B.11 Adjusts the format of the exon columns, in normal and tumor files, to be in a dataframe list datatype format (Python, Pycharm)



```
######### fixing the format of the Exon Lists from refFlat files ############
for row in range(norm.shape[0]):
    #Changing Exon Starts from string to list
    norm["exonStarts"][row] = norm["exonStarts"][row].split(",")
    norm["exonStarts"][row] = norm["exonStarts"][row][:-1]

    norm["exonStarts"][row] = [eval(i) for i in norm["exonStarts"][row]]

    # Changing Exon Ends from string to list
    norm["exonEnds"][row] = norm["exonEnds"][row].split(",")
    norm["exonEnds"][row] = norm["exonEnds"][row][:-1]

    norm["exonEnds"][row] = [eval(i) for i in norm["exonEnds"][row]]

for row in range(tumor.shape[0]):
    #Changing Exon Starts from string to list
    tumor["exonStarts"][row] = tumor["exonStarts"][row].split(",")
    tumor["exonStarts"][row] = tumor["exonStarts"][row][:-1]

    tumor["exonStarts"][row] = [eval(i) for i in tumor["exonStarts"][row]]

    #Changing Exon Ends from string to list
    tumor["exonEnds"][row] = tumor["exonEnds"][row].split(",")
    tumor["exonEnds"][row] = tumor["exonEnds"][row][:-1]

    tumor["exonEnds"][row] = [eval(i) for i in tumor["exonEnds"][row]]

norm.to_csv("Normal_refFlat.csv", sep=",")
tumor.to_csv("Tumor_refFlat.csv", sep=",")
```

B.12 Inserts five additional columns, for each of the four bases (A, G, T, C) and the full sequence. The sequence is obtained from a .fasta file with the full chromosome sequence (using gene's exon columns to parse this into the desired segment) and the four base columns are the counts in which they occur in the found sequence (Python, Pycharm).

```
sequences = []
for row in range(UN.shape[0]):
    chr = UN.iloc[row]["chrom"]
    if UN.iloc[row]["Length"] == 0:
        sequences.append(0)
        continue
    start = int(UN.iloc[row]["txStart"])
    end = int(UN.iloc[row]["txEnd"])
    strand = UN.iloc[row]["strand"]

    temp = geneSeq(chr, start, end, strand)
    UN.loc[row, 'Sequence'] = temp
    UN.loc[row, 'A'] = temp.count("A")
    UN.loc[row, 'G'] = temp.count("G")
    UN.loc[row, 'T'] = temp.count("T")
    UN.loc[row, 'C'] = temp.count("C")
#
# UN = UN.insert(UN.shape[0], "Sequence", sequences)
UN.to_csv("check1Normal_A*L.csv", index = False)

for row in range(UT.shape[0]):
    chr = UT.iloc[row]["chrom"]
    if UT.iloc[row]["Length"] == 0:
        sequences.append(0)
        continue
    start = int(UT.iloc[row]["txStart"])
    end = int(UT.iloc[row]["txEnd"])
    strand = UT.iloc[row]["strand"]

    temp = geneSeq(chr, start, end, strand)
    UT.loc[row, 'Sequence'] = temp
    UT.loc[row, 'A'] = temp.count("A")
    UT.loc[row, 'G'] = temp.count("G")
    UT.loc[row, 'T'] = temp.count("T")
    UT.loc[row, 'C'] = temp.count("C")
#
# UN = UN.insert(UN.shape[0], "Sequence", sequences)
UT.to_csv("check2Tumor_A*L.csv", index = False)
```

B.13 Calculates the length of the sequence, inserts this value in a new column (Python, Pycharm).

```
######### Get lengths of sequences ######
length = []
for row in range(uniqueN.shape[0]):
    if uniqueN.iloc[row]["txStart"] == 0:
        length.append(0)
        continue
    min = int(uniqueN.iloc[row]["txStart"])
    max = int(uniqueN.iloc[row]['txEnd'])
    temp = max - min
    length.append(temp)
uniqueN = uniqueN.insert(uniqueN.shape[1], "Length", length)
```

B.14 Function that sums the four base columns, returns this value (Python, Pycharm).

```
def counts(csv):
    a = csv["A"].sum()
    g = csv["G"].sum()
    t = csv["T"].sum()
    c = csv["C"].sum()
    print("A: " + str(a))
    print("G: " + str(g))
    print("T: " + str(t))
    print("C: " + str(c))

print("Normal:")
counts(n)
print("Tumor:")
counts(t)
```

B.15 Using the unique gene values, in the cases that have many isoforms, the isoform with the largest length is selected and the others are dropped (Python, Pycharm).

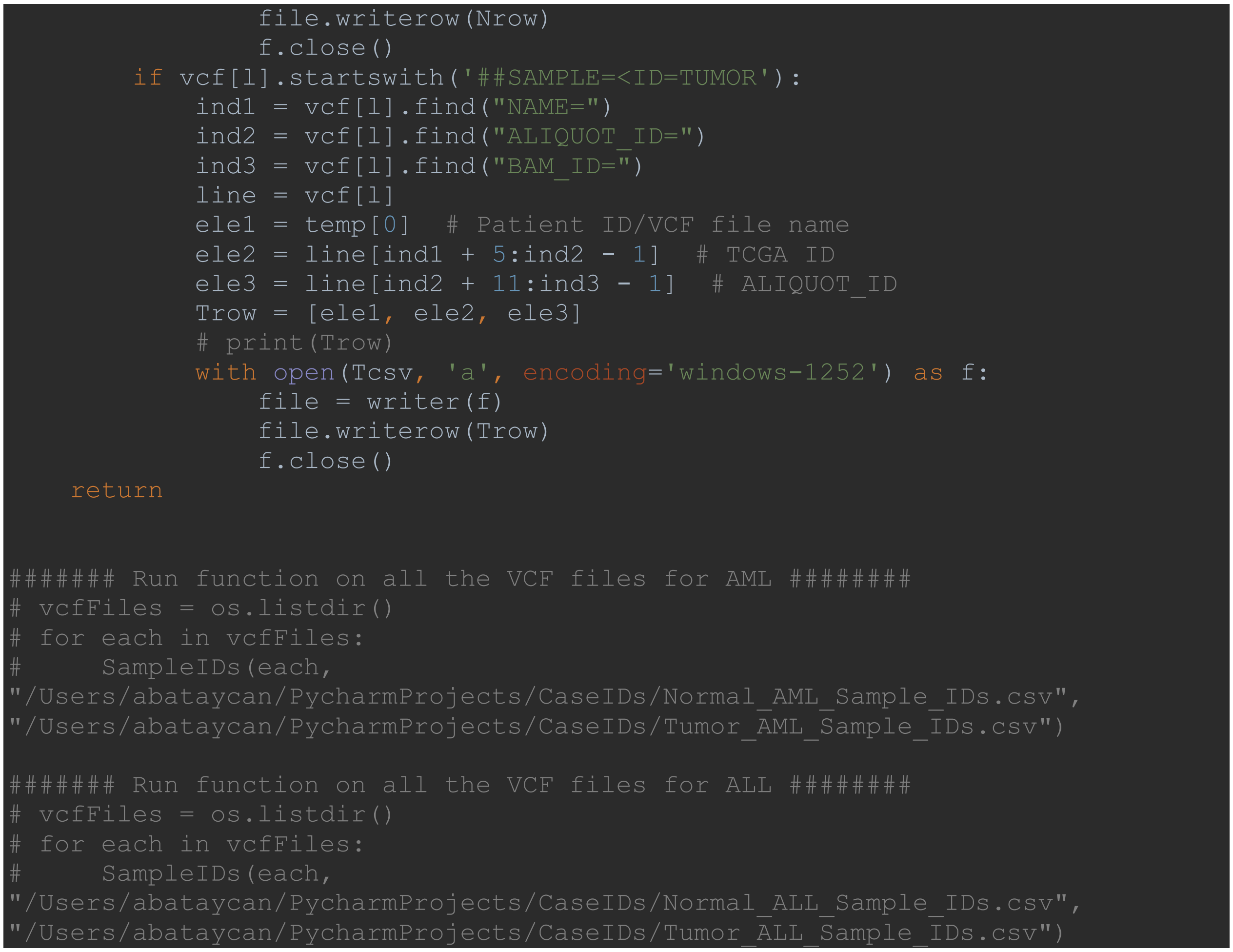
```
##################### Use only the unique genes to avoid double counting,
isoforms present by taking the largest length of the same gene, strand, and
chrom ##########################
########## ALL ###########
UN = pd.read_csv("ALL_Normal_UniqueGenes.csv", sep=',')
UT = pd.read_csv("ALL_Tumor_UniqueGenes.csv", sep=',')
UN = UN.sort_values(by="Length", ascending=False).drop_duplicates(['chrom',
'strand', "New_Gene_Name"]).sort_index()
UT = UT.sort_values(by="Length", ascending=False).drop_duplicates(['chrom',
'strand', "New_Gene_Name"]).sort_index()
UN.to_csv("ALL_Normal_UniqueGenes.csv", index = False, sep=',')
UT.to_csv("ALL_Tumor_UniqueGenes.csv", index = False, sep=',')
########## AML ##############
UN = pd.read_csv("AML_Normal_UniqueGenes.csv", sep=',')
UT = pd.read_csv("AML_Tumor_UniqueGenes.csv", sep=',')
UN = UN.sort_values(by="Length", ascending=False).drop_duplicates(['chrom',
'strand', "New_Gene_Name"]).sort_index()
UT = UT.sort_values(by="Length", ascending=False).drop_duplicates(['chrom',
'strand', "New_Gene_Name"]).sort_index()
UN.to_csv("AML_Normal_UniqueGenes.csv", index = False, sep=',')
UT.to_csv("AML_Tumor_UniqueGenes.csv", index = False, sep=',')
```

B.16 Scores each gene, based on the FATHMM results, inserts these scores into a new column (Python, Pycharm).

```
def geneWeights(csv, file):
    for row in range(csv.shape[0]):
        genelen = csv.loc[row, "Length"]
        sum = 0.0

        #clean Score column and convert string to list
        score = csv.loc[row, "Tumor Total Score"].replace("[", "")
        score = score.replace("'", "")
        score = score.replace("]", "")
        score = score.replace(" ", "")
        score = score.split(",")

        #clean tumor count and normal count and convert string to list
        tumor = csv.loc[row, "SNV_Tumor_Total"].replace("[", "")
        tumor = tumor.replace("'", "")
        tumor = tumor.replace("]", "")
        tumor = tumor.replace(" ", "")
        tumor = tumor.split(",")

        normal = csv.loc[row, "SNV_Normal_Total"].replace("[", "")
        normal = normal.replace("'", "")
        normal = normal.replace("]", "")
        normal = normal.replace(" ", "")
        normal = normal.split(",")

        for (a, b, c) in zip(score, tumor, normal):
            diff = int(b) - int(c)
            temp = float(a) * float(diff)
            sum += temp
        if genelen == 0:
            co = 0
        else:
            co = 1.0 / (ln(genelen))

        val = co*sum

        csv.loc[row, "Gene_Score"] = val


    csv.drop(csv[csv['New_Gene_Name'] == "NoName"].index, inplace=True)

    csv.to_csv(file, sep=",", index=False)
```

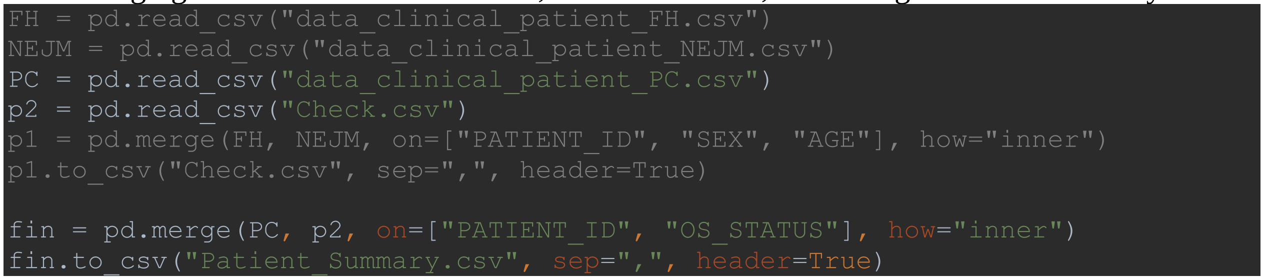
B.17 Reads in a folder of VCF file and extracts the sample/patient information.

```
def CaseIDs(vcffile, csv):
    temp = vcffile.split(".")
    vcffile = open('%s'%(vcffile), 'r')
    vcf = vcffile.readlines()
    vcffile.close()

    for l in range(0, len(vcf)): #looking line by line in the inputted VCF
file
        if vcf[l].startswith('##INDIVIDUAL'):
            ind = vcf[l].find("ID=")
            line = vcf[l]
            if ind == -1:
                print("Problem with: " + temp[0])
                return
            else:
                #print(line[ind+3:-2])
                ele1 = temp[0] #Patient ID/VCF file name
                ele2 = line[ind+3:-2] #Case ID that will be matched with cBio
portal
                row = [ele1, ele2]
                #print(row)
                with open(csv, 'a', encoding='windows-1252') as f:
                    file = writer(f)
                    file.writerow(row)
                    f.close()

    return


####### Run function on all the VCF files for AML ########
# vcfFiles = os.listdir()
# for each in vcfFiles:
#     CaseIDs(each,
"/Users/abataycan/PycharmProjects/CaseIDs/CaseIDS_ALL.csv")




####### Run function on all the VCF files for ALL ########
# vcfFiles = os.listdir()
# for each in vcfFiles:
#     CaseIDs(each,
"/Users/abataycan/PycharmProjects/CaseIDs/CaseIDS_ALL.csv")



def SampleIDs(vcffile, Ncsv, Tcsv):
    temp = vcffile.split(".")
    vcffile = open('%s'%(vcffile), 'r')
    vcf = vcffile.readlines()
    vcffile.close()

    for l in range(0, len(vcf)): #looking line by line in the inputted VCF
file
        if vcf[l].startswith('##SAMPLE=<ID=NORMAL'):
            ind1 = vcf[l].find("NAME=")
            ind2 = vcf[l].find("ALIQUOT_ID=")
            ind3 = vcf[l].find("BAM_ID=")
            line = vcf[l]
            #print(line[ind+3:-2])
            ele1 = temp[0] #Patient ID/VCF file name
            ele2 = line[ind1+5:ind2-1] #TCGA ID
            ele3 = line[ind2+11:ind3-1] #ALIQUOT_ID
            Nrow = [ele1, ele2, ele3]
            # print(Nrow) #normal sample assorted ids
            with open(Ncsv, 'a', encoding='windows-1252') as f:
                file = writer(f)
```



```
                file.writerow(Nrow)
                f.close()
        if vcf[l].startswith('##SAMPLE=<ID=TUMOR'):
            ind1 = vcf[l].find("NAME=")
            ind2 = vcf[l].find("ALIQUOT_ID=")
            ind3 = vcf[l].find("BAM_ID=")
            line = vcf[l]
            ele1 = temp[0]  # Patient ID/VCF file name
            ele2 = line[ind1 + 5:ind2 - 1]  # TCGA ID
            ele3 = line[ind2 + 11:ind3 - 1]  # ALIQUOT_ID
            Trow = [ele1, ele2, ele3]
            # print(Trow)
            with open(Tcsv, 'a', encoding='windows-1252') as f:
                file = writer(f)
                file.writerow(Trow)
                f.close()
    return


####### Run function on all the VCF files for AML ########
# vcfFiles = os.listdir()
# for each in vcfFiles:
#     SampleIDs(each,
"/Users/abataycan/PycharmProjects/CaseIDs/Normal_AML_Sample_IDs.csv",
"/Users/abataycan/PycharmProjects/CaseIDs/Tumor_AML_Sample_IDs.csv")

####### Run function on all the VCF files for ALL ########
# vcfFiles = os.listdir()
# for each in vcfFiles:
#     SampleIDs(each,
"/Users/abataycan/PycharmProjects/CaseIDs/Normal_ALL_Sample_IDs.csv",
"/Users/abataycan/PycharmProjects/CaseIDs/Tumor_ALL_Sample_IDs.csv")
```

B.18 Merging the three clinical databases, from cBioPortal, into a single clinical summary.



```
FH = pd.read_csv("data_clinical_patient_FH.csv")
NEJM = pd.read_csv("data_clinical_patient_NEJM.csv")
PC = pd.read_csv("data_clinical_patient_PC.csv")
p2 = pd.read_csv("Check.csv")
p1 = pd.merge(FH, NEJM, on=["PATIENT_ID", "SEX", "AGE"], how="inner")
p1.to_csv("Check.csv", sep=",", header=True)

fin = pd.merge(PC, p2, on=["PATIENT_ID", "OS_STATUS"], how="inner")
fin.to_csv("Patient_Summary.csv", sep=",", header=True)
```

APPENDIX C

C.1 Compares the unique patients in tumor file to the normal file, identifies the ID of those who are missing (R, Rstudio).

~~~~
## Checking for the missing patients between normal and tumor samples...
```{r}
        tum <- read.csv("AML_Tumor.csv", header=TRUE, sep=",")
        length(table(tum$Patient_ID))
        norma <- read.csv("AML_Normal.csv", header=TRUE, sep=",")
        length(table(norma$Patient_ID))

        unique(norma$Patient_ID)[!unique(norma$Patient_ID) %in% unique(tum$Patient_ID)]
        unique(tum$Patient_ID)[!unique(tum$Patient_ID) %in% unique(norma$Patient_ID)]

        tum <- read.csv("ALL_Tumor.csv", header=TRUE, sep=",")
        length(table(tum$Patient_ID))
        norma <- read.csv("ALL_Normal.csv", header=TRUE, sep=",")
        length(table(norma$Patient_ID))

        unique(norma$Patient_ID)[!unique(norma$Patient_ID) %in% unique(tum$Patient_ID)]
        unique(tum$Patient_ID)[!unique(tum$Patient_ID) %in% unique(norma$Patient_ID)]
```
~~~~

C.2 Calculates the variant densities (per Megabase) for each chromosomes, plots the values on a bar plot sort by chromosome, and generates a table with these values with the tumor to normal ratio (R, Rstudio).

~~~~
## Chromosome density Barplot
```{r}
len = c(248956422, 242193529, 198295559, 190214555, 181538259, 170805979, 159345973,
145138636, 138394717, 133797422, 135086622, 133275309, 114364328, 107043718,
~~~~

~~~
101991189, 90338345, 83257441, 80373285, 58617616, 64444167, 46709983, 156040895,
57227415)

        order = c("chr1", "chr2", "chr3", "chr4", "chr5", "chr6", "chr7", "chr8", "chr9", "chr10",
"chr11", "chr12", "chr13", "chr14", "chr15", "chr16", "chr17", "chr18", "chr19", "chr20", "chr21",
"chrX", "chrY")

        tChromFreq <- table(tumor$chrom)
        t = (tChromFreq[order]/len)*1000000
        barplot(t, col = rainbow(length(tChromFreq[order])), main = "AML Tumor Variants per
Mega-base (Mb)")

        nChromFreq <- table(normal$chrom)
        n = (nChromFreq[order]/len)*1000000
        barplot(n, col = rainbow(length(nChromFreq[order])), main = "AML Normal Variants per
Mega-base (Mb)")

        tab = data.frame(Variants=tChromFreq[order], Length=len, "Normalize(1000)"=t)
        tab = tab[ , c(-4)]

        tabn = data.frame(Variants=nChromFreq[order], Length=len, "Normalize(1000)"=n)
        tabn = tabn[ , c(-4)]
```
~~~

C.3 Using the clinical data, creates a table with the number of patients for there current status, gender, ethnicity for each stage of AML. Generates a notched boxplot based on their current status comparing the overall survival in month against the stage classification (R, Rstudio).

~~~
## FAB Notched Boxplot and identifying others values in demographic csv
```{r}
	#sub_Cl <- read.csv("sub_AML_Clinicals.csv", header=TRUE, sep = ",")
	sub_Cl <- read.csv("subset_AML_FAB.csv", header=TRUE, sep = ",")

	boxplot(OS_MONTHS~FAB, data = sub_Cl, notch = TRUE, col=rainbow(length(table(sub_Cl$FAB))), main = "French, American, and British Classification against Overall Survival", ylab = "Overall Survival (in Months)", xlab = "FAB Classification")

	# prints table of the unique values found in RACE and SEX columns from the demographic data
	table(sub_Cl$RACE)
	table(sub_Cl$SEX)

	X <- split(sub_Cl, sub_Cl$OS_STATUS)
	living <- X$`0:LIVING`
	deceased <- X$`1:DECEASED`
	table(living$FAB)
	table(deceased$FAB)

	graph_living <- boxplot(OS_MONTHS~FAB, data = living, notch = TRUE, col=rainbow(length(table(sub_Cl$FAB))), main = "French, American, and British Classification against Overall Survival (Living)", ylab = "Overall Survival (in Months)", xlab = "FAB Classification")

	graph_deceased <- boxplot(OS_MONTHS~FAB, data = deceased, notch = TRUE, col=rainbow(length(table(sub_Cl$FAB))), main = "French, American, and British Classification
~~~

~~~
against Overall Survival (Deceased)", ylab = "Overall Survival (in Months)", xlab = "FAB
Classification")
```
~~~

Vita

Amanda Bataycan was born July 27th, 1994, in Rhode Island. Belonging to a blended family with five kids, with such a large family she learned the importance of teamwork and leadership skills from which her parents displayed. Beginning her studies at New Mexico State University, she obtained Bachelor of Science degrees in Applied Mathematics and Biology with a minor in Chemistry by 2016. Prior to graduation, she started working as a private tutor, specializing in STEM related subjects, for students ranging from elementary to high school levels. Before returning to school, she became a private pilot and was elected as the Treasurer of the Ninety Nines South Central Section in El Paso Chapter. While working at an endocrinology clinic, in 2019, she joined the Bioinformatic Program at the University of Texas at El Paso (UTEP) and received her master's in 2021. At this time, she also decided to continue her studies in the Computational Science Program at UTEP. She continues to work at the clinic as an Operational Manager and has also been working as a Teaching Assistant in the Mathematical Sciences Department, with the primary focus that studies come first. In 2022, she was elected as the Secretary of the Computational Science Student Association, allowing her to connect with first-year students in her program and provide guidance with a well understand approach of the struggles they face.