**Methods**

**Code Used for Database Creation**

**Merge VCF Files**

#In PuTTY terminal

#move into directory with VCF files

cd cohort\_combined-vcf

#list contents of folder

ls

#activate environment

Conda activate bioinfo

#using bcftools, create an ID column and output file

bcftools annotate --set-id '%CHROM:%POS:%REF:%FIRST\_ALT' merged-vars.vcf.gz -Oz -o merged-vars-ID-class.vcf.gz

**Annotate VCF Files**

The following parameters were used in the wANNOVAR annotator

* Input File: merged-vars-ID-class.vcf.gz
* Result Duration: 1 day
* Reference Duration: hg19
* Input Format: VCF
* Gene Definition: RefSeq Gene
* Individual Analysis: All annotation
* Disease Model: none

**Using Plink**

#In PuTTY terminal

#activate PLINK environment

Conda activate plink

#run PLINK for genotype count report

plink --vcf merged-vars-ID-class.vcf.gz --freqx --out merged-ID-order --keep-allele-order

#run Plink for minor allele frequency

plink --vcf merged-vars-ID-class.vcf.gz --freq --out merged-ID-order --keep-allele-order

**Formatting Annotated Excel Sheet**

In the main tab

* Set the gnomad link
* Add a leftmost column called “gnomad”
* Add a link for each variant using the formula below

#formula for gnomAD link

=HYPERLINK(CONCAT("https://gnomad.broadinstitute.org/variant/",C2,"-",D2,"-",E2,"-",F2,"?dataset=gnomad\_r2\_1"),"gnomad")

Including PLINK results

* Add 7 new columns to the right of the amino acid change tab
* Name the new column 1 “C(HOM A1)”
* Enter the formula: =VLOOKUP([@SNPid],PLINK!$B:$L,4,FALSE)
* Name the new column 2 “C(HET)”
* Enter the formula: =VLOOKUP([@SNPid],PLINK!$B:$L,5,FALSE)
* Name the new column 3 “C(HOM A2)”
* Enter the formula: =VLOOKUP([@SNPid],PLINK!$B:$L,6,FALSE)
* Name the new column 4 “MAF\_PLINK”
* Enter the formula: =VLOOKUP([@SNPid],PLINK!$B:$L,10,FALSE)
* Name the new column 5 “MAF”
* Calculate the minor allele frequency by hand by dividing C(HET) by 420 (total alleles)
* Enter the formula: N2/420
* Name the new column 6 “MAF\_%”
* Calculate the minor allele frequency as a percent
* Enter the formula: P2\*100
* Name the new column 7 “NCHROBS”
* Enter the formula: =VLOOKUP([@SNPid],PLINK!$B:$L,11,FALSE)

**Code Used for Exploratory Data Analysis**

**Load in Libraries and Data**

#load in library

library(tidyverse) #loads ggplot

library(infer) #loads tidyverse

#load in csv file

library(readxl)

data <- read\_xlsx("D:/!Capstone(Summer2025)/project files/ANNOVAR\_19\_full\_annotation-class.xlsx")

**Data Cleaning and Manipulation**

#list variables in dataset

ls(data)

#rename variables to remove spaces and parenthesis

data=data %>%

rename(HOM\_A1= `C(HOM A1)`)

data=data %>%

rename(HOM\_A2=`C(HOM A2)`)

data=data %>%

rename(HET=`C(HET)`)

data=data %>%

rename(MAF\_percent=`MAF\_%`)

#list of variables and their types

str(data)

#Check counts for Categorical Variables

data %>%

count(Chr)

data %>%

count(Func)

data %>%

count(Gene)

data %>%

count(ExonicFunc)

data %>%

count(HOM\_A1)

data %>%

count(HOM\_A2)

data %>%

count(HET)

data%>%

count(SIFT\_pred)

data%>%

count(LRT\_pred)

data%>%

count(MutationTaster\_pred)

data%>%

count(MutationAssessor\_pred)

data %>%

count(FATHMM\_pred)

#For ExonicFunc,SIFT\_pred,LRT\_pred,MutationTaster\_pred,MutationAssessor\_pred,FATHMM\_pred variables, change "." to NA

data=data %>%

mutate(ExonicFunc=ifelse(ExonicFunc==".", NA, ExonicFunc))

data=data %>%

mutate(SIFT\_pred=ifelse(SIFT\_pred==".", NA, SIFT\_pred))

data=data %>%

mutate(LRT\_pred=ifelse(LRT\_pred==".", NA, LRT\_pred))

data=data %>%

mutate(MutationTaster\_pred=ifelse(MutationTaster\_pred==".", NA, MutationTaster\_pred))

data=data %>%

mutate(MutationAssessor\_pred=ifelse(MutationAssessor\_pred==".", NA, MutationAssessor\_pred))

data=data %>%

mutate(FATHMM\_pred=ifelse(FATHMM\_pred==".", NA, FATHMM\_pred))

#check correction

data %>%

count(ExonicFunc)

data %>%

count(SIFT\_pred)

data %>%

count(LRT\_pred)

data %>%

count(MutationTaster\_pred)

data %>%

count(MutationAssessor\_pred)

data %>%

count(FATHMM\_pred)

#Summary for quantitative variables

data %>%

summary(data)

#look at quantitative variables

data %>%

count(MAF\_PLINK)

data %>%

count(MAF)

data %>%

count(MAF\_percent)

data %>%

count(NCHROBS)

data %>%

count(AF)

data %>%

count(AF\_nfe)

data %>%

count(REVEL\_score)

data %>%

count(CADD\_raw)

data %>%

count(`GERP++\_NR`)

data %>%

count(`GERP++\_RS`)

#For AF,AF\_nfe,REVEL\_score,GERP++\_NR variables, change "." to NA

data=data %>%

mutate(AF=ifelse(AF==".", NA, AF))

data=data %>%

mutate(AF\_nfe=ifelse(AF\_nfe==".", NA, AF\_nfe))

data=data %>%

mutate(REVEL\_score=ifelse(REVEL\_score==".", NA, REVEL\_score))

data=data %>%

mutate(`GERP++\_NR`=ifelse(`GERP++\_NR`==".", NA, `GERP++\_NR`))

#check correction

data %>%

count(AF)

data %>%

count(AF\_nfe)

data %>%

count(REVEL\_score)

data %>%

count(`GERP++\_NR`)

#remove all NA's in dataframe

data <- na.omit(data)

**Variation**

#Chromosome Bar Chart

data %>%

ggplot()+geom\_bar(aes(x=Chr))+labs(title="Chromosome", x="Chr", y="count")

#Function Bar Chart

data %>%

ggplot()+geom\_bar(aes(x=Func))+labs(title="Function", x="Func", y="count")

#Exonic Function Bar Chart

data %>%

ggplot()+geom\_bar(aes(x=ExonicFunc))+labs(title="Exonic Function", x="Exonic Function", y="count")

#SIFT Predictor Bar Chart

data %>%

ggplot()+geom\_bar(aes(x=SIFT\_pred))+labs(title="SIFT Predictor", x="SIFT\_pred", y="count")

#LRT Predictor Bar Chart

data %>%

ggplot()+geom\_bar(aes(x=LRT\_pred))+labs(title="LRT Predictor", x="LRT\_pred", y="count")

#Mutation Taster Predictor Bar Chart

data %>%

ggplot()+geom\_bar(aes(x=MutationTaster\_pred))+labs(title="MutationTaster Predictor", x="MutationTaster\_pred", y="count")

#Mutation Assessor Predictor Bar Chart

data %>%

ggplot()+geom\_bar(aes(x=MutationAssessor\_pred))+labs(title="MutationAssessor Predictor", x="Mutation Assessor\_pred", y="count")

#FATHMM Predictor Bar Chart

data %>%

ggplot()+geom\_bar(aes(x=FATHMM\_pred))+labs(title="FATHMM Predictor", x="FATHMM\_pred", y="count")

**Covariation**

#Gene Function vs Exonic Function Bar Chart

data %>%

ggplot()+geom\_bar(aes(x=Func, fill=ExonicFunc))+labs(title="Gene Function vs Exonic Function", x="Function", y="count")

#Chromosome Number vs Function

data %>%

ggplot()+geom\_bar(aes(x=Chr, fill=Func))+labs(title="Chromosome Number vs Function", x="Chromosome", y="count")

#Chromosome Number vs Exonic Function

data %>%

ggplot()+geom\_bar(aes(x=Chr, fill=ExonicFunc))+labs(title="Chromosome Number vs Exonic Function", x="Chromosome", y="count")

#Chromosome Number vs SIFT Predictor

data %>%

ggplot()+geom\_bar(aes(x=Chr, fill=SIFT\_pred))+labs(title="Chromosome Number vs SIFT Predictor", x="Chromosome", y="count")

#Chromosome Number vs LRT Predictor

data %>%

ggplot()+geom\_bar(aes(x=Chr, fill=LRT\_pred))+labs(title="Chromosome Number vs LRT Predictor", x="Chromsome", y="count")

#Chromosome Number vs MutationTaster Predictor",

data %>%

ggplot()+geom\_bar(aes(x=Chr, fill=MutationTaster\_pred))+labs(title="Chromosome Number vs MutationTaster Predictor", x="Chromosome", y="count")

#Chromosome Number vs MutationAssessor Predictor

data %>%

ggplot()+geom\_bar(aes(x=Chr, fill=MutationAssessor\_pred))+labs(title="Chromosome Number vs MutationAssessor Predictor", x="Chromosome", y="count")

#Chromosome Number vs FATHMM Predictor

data %>%

ggplot()+geom\_bar(aes(x=Chr, fill=FATHMM\_pred))+labs(title="Chromosome Number vs FATHMM Predictor", x="Chromosome", y="count")

**Genes of Interest**

GOI\_data=data %>%

filter(Gene %in% c("APOB","PCCB","PAH","ZDHHC16","RYR2","BCKDHA","GCDH","PKLR","ACADM","SLC12A3","FLVCR1","MTHFR"))

GOI\_data=GOI\_data %>%

select(-gnomad,-GeneDetail,-rsNum)

#Genes of Interest vs Chromosome

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene))+facet\_wrap(~Chr)+labs(title="Genes of Interest on each Chromosome", x="Genes of Interest", y="count")+theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

#Genes of Interest vs Predictor

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene, fill=SIFT\_pred))+labs(title="SIFT Predictor", x="Gene", y="count")+theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene, fill=LRT\_pred))+labs(title="LRT Predictor", x="Gene", y="count")+theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene, fill=MutationTaster\_pred))+labs(title="MutationTaster Predictor", x="Gene", y="count")+theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene, fill=MutationAssessor\_pred))+labs(title="MutationAssessor Predictor", x="Gene", y="count")+theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene, fill=FATHMM\_pred))+labs(title="FATHMM Predictor", x="Gene", y="count")+theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

#Genes of Interest vs Chromosome with Predictor

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene,fill=SIFT\_pred))+facet\_wrap(~Chr)+labs(title="Genes of Interest on each Chromosome", x="Genes of Interest", y="count")+theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene,fill=LRT\_pred))+facet\_wrap(~Chr)+labs(title="Genes of Interest on each Chromosome", x="Genes of Interest", y="count")+theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene,fill=MutationTaster\_pred))+facet\_wrap(~Chr)+labs(title="Genes of Interest on each Chromosome", x="Genes of Interest", y="count")+ theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene,fill=MutationAssessor\_pred))+facet\_wrap(~Chr)+labs(title="Genes of Interest on each Chromosome", x="Genes of Interest", y="count")+theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

GOI\_data %>%

ggplot()+geom\_bar(aes(x=Gene,fill=FATHMM\_pred))+facet\_wrap(~Chr)+labs(title="Genes of Interest on each Chromosome", x="Genes of Interest", y="count")+theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))

#Filter for only distinctly harmful genes

GOI\_Harmful\_data=GOI\_data %>%

filter(SIFT\_pred=="D") %>%

filter(LRT\_pred=="D") %>%

filter(MutationTaster\_pred=="D") %>%

filter(MutationAssessor\_pred=="M") %>%

filter(FATHMM\_pred=="D")

#list of all harmful Genes of Interest

GOI\_Harmful\_data %>%

count(Gene)

**Filter for Harmful Genes**

#Filter for only distinctly harmful genes

Harmful\_data=data %>%

filter(SIFT\_pred=="D") %>%

filter(LRT\_pred=="D") %>%

filter(MutationTaster\_pred=="D") %>%

filter(MutationAssessor\_pred=="M") %>%

filter(FATHMM\_pred=="D")

#list all harmful genes in whole database

Harmful\_data %>%

count(Gene)

#Filter for homozygous for Allele 1 and harmful genes

Harmful\_data %>%

filter(HOM\_A1>0) %>%

count(Gene,Chr,MAF\_percent,REVEL\_score,CADD\_raw,`GERP++\_NR`) %>%

arrange(desc(MAF\_percent))

#Filter for homozygous for Allele 2 and harmful genes

Harmful\_data %>%

filter(HOM\_A2>0) %>%

count(Gene)

#Filter for heterozygous and harmful genes

Harmful\_data %>%

filter(HET>0) %>%

count(Gene,Chr,MAF\_percent,REVEL\_score,CADD\_raw,`GERP++\_NR`) %>%

arrange(desc(MAF\_percent))