**Using Cancer Effect Size to Quantify Driver Mutation Impact in Chinese Hepatocellular Carcinoma Genomes**

By Ruiheng Wu

Date: May 2, 2025

Course Number: BIS 681

I have not given, received, or witnessed inappropriate exchange of information on this assignment, and I certify that this is my own original work.

**Abstract**

Hepatocellular carcinoma (HCC) is a leading cause of cancer-related mortality worldwide, with significant geographic and genetic heterogeneity. The Chinese Liver Cancer Atlas (CLCA) project performed deep whole genome sequencing (WGS) of 494 HCC tumors, providing a rich resource for characterizing the landscape of somatic mutations in a Chinese population. In this study, we re-analyzed the CLCA WGS data using the Townsend Lab’s cancer effect size framework to quantify the selection intensities of individual mutations. This approach highlights that not all frequently mutated genes exert equal selective pressure in tumor evolution. We identified several candidate driver mutations under strong positive selection, including recurrent variants in CTNNB1, TP53, and JAK1, along with a previously underreported mutation in P4HA1. To assess cross-cohort commonality, we compared selection strengths between CLCA-HCC and The Cancer Genome Atlas Liver Hepatocellular Carcinoma project (TCGA‑LIHC), an American-enriched cohort, and found substantial overlap in high-effect variants, particularly in CTNNB1 and TP53. Furthermore, we investigated epistatic interactions among top driver genes and discovered significant variant-level epistasis between JAK1 and CTNNB1, suggesting potential synergistic effects in tumorigenesis. Lastly, we performed survival analysis using TCGA-LIHC clinical data. Kaplan-Meier and log-rank tests confirmed that TP53 mutations are significantly associated with poor overall survival, underscoring its clinical importance. Together, our results demonstrate the power of integrating cancer effect size modeling with epistasis analysis and survival outcomes to uncover biologically meaningful and clinically relevant drivers in HCC.

**Introduction**

Hepatocellular carcinoma (HCC) is the most prevalent form of primary liver cancer and ranks as the third leading cause of cancer-related deaths globally, with especially high incidence rates in East Asia [1]. Therefore, genetic diagnosis for patients with HCC would be important for understanding cancer development. The whole genome sequencing (WGS) can reveal individual genetic predispositions through identifying somatic mutations in cancer cells, guiding treatment decisions and potentially leading to targeted therapies [2]. The Chinese Liver Cancer Atlas (CLCA) project recently performed deep WGS on 494 HCC tumors from Chinese patients, uncovering numerous novel coding and non-coding driver candidates [3].

The identification of driver mutations, which confer a selective advantage to tumor cells, is central to understanding cancer development and informing targeted therapy. Current study on CLCA-HCC identified candidate driver genes and their mutations through the Fisher’s exact test [3]. Historically, many cancer genomics studies used similar strategies to rank potential driver genes using mutation prevalence or statistical significance (P values) of mutation enrichment [4]. However, these metrics are confounded by gene-specific mutation rates, potentially elevating biologically implausible genes to top ranks due to sheer length and background mutation bias, rather than functional relevance. To overcome these limitations, modeling frameworks have emerged to estimate the “cancer effect size” as a measure of the selective advantage conferred by a somatic variant on tumor growth by integrating observed mutation prevalence with site-specific baseline mutation rates [5]. In this study, we apply this framework to the CLCA dataset, which comprises deep WGS data of HCC tumors [3]. Our goals are to: (1) identify mutations under strong positive selection in Chinese HCC; (2) compare selective pressures across Chinese enriched (CLCA-HCC) and American enriched (TCGA-LIHC) cohorts; (3) evaluate epistatic interactions among top mutations; and (4) assess the potential significance of key genes (*TP53*, *CTNNB1*, *JAK1*) in relation to overall survival. We hypothesize that highly prevalent mutations are not uniformly impactful, that variant-level epistasis shapes tumor evolution, and that selective pressure metrics offer biologically and clinically meaningful insights beyond conventional mutation frequency analyses.

**Study Methods and Procedures**

We started with identification of candidate driver genes and mutants through analyzing the Mutation Annotation Format (MAF) files that containing important somatic variant information, like the occurrence of single‑nucleotide variants (SNVs) and small indels in the “cancereffectsizeR” pipeline. MAF data are based on whole‑genome sequencing data with reference genome of hg19/GRCh37 from 494 HCC tumors enrolled in the CLCA study. The MAF data based on whole exome sequencing (WES) data with reference genome of hg38/GRCh38 from 357 tumors TCGA-LIHC are available through Genomic Data Commons of National Cancer Institute.

After identifying the top genes and mutants having strong selection strength in cancer, the overall survival of patients in TCGA‑LIHC cohort were analyzed using the Kaplan–Meier method. Through stratifying patients by mutation status of selected genes, differences between curves were assessed with two‑sided log‑rank tests.

**Bioinformatics Pipeline and Statistical Analysis**

In general, the “cancereffectsizeR” pipeline combined the mutation information at site-level and gene-level, then infer their selection in tumor as the “effect sizes”.

1. Mutation rate extraction

The trinucleotide‑context specific mutation rates are derived from COSMIC mutational signature extractions [6]. In this process, the relative frequencies of each possible nucleotide‑change in DNA (e.g., C>T in an A[C]G triplet) is detected and matched for signatures through COSMIC database. The analysis would inform the frequencies of particular DNA nucleotides and biological context in the HCC tumors.

1. Selection Inferences

The selection of variant and gene is based on the dNdScv model [7]. As million combinations of DNA nucleotides coded to 20 amino acids in triplet codons, we would be able to determine the silent synonymous mutations (S) and non-synonymous mutations (N) for each variant mutation. Because synonymous changes rarely affect protein function, they are assumed to be selectively neutral and thus reflect the underlying mutation process instead of selection. The number of synonymous () and non-synonymous () mutations follows a Poisson framework [equation 1].

**Equation 1**: Poisson framework for selection inference.

Therefore, the change of reflecting the selection, and is the trinucleotide mutation rate calculated before. Gene‑level effects obtained by aggregating non-synonymous variants for each gene and refitting.

1. Pairwise Epistasis Analysis

Epistasis analysis is the established to detect the shift of selection of variant/gene A with/without the presence of variant/gene B. Any pair of variants/gene resulted in P value < 0.05 after likelihood-ratio test (LRT) indicating significant epistasis relationship.

1. Survival analysis

Overall survival in TCGA‑LIHC is analyzed with Kaplan–Meier curves and two‑sided log‑rank tests [equation 2].

**Equation 2:** Surivival Probability function and log-rank test statistics function. is survival probability estimation at each time point. is number of people living at the start. is number of death util each time point. is the sum of observed events and is the sum of expected events

**Results**:

The pipeline identified CTNNB1, TP53, and JAK1 as having the highest scaled selection coefficients (>100,000) in the CLCA‑HCC cohort, confirming their dominant evolutionary roles in hepatocellular carcinoma and corroborating previous findings [3]. By contrast, **P4HA1** emerged as a novel high‑effect driver. Despite its low prevalence, the D372A variant displayed an exceptionally large selection coefficient (>10,000,000).

Cross‑cohort analysis with the same pipeline applied to TCGA‑LIHC revealed five shared driver genes (CTNNB1, TP53, JAK1, NFIC, and PIK3CA) encompassing 18 recurrent variants. CTNNB1 accounted for 11 of these variants, underscoring its universal importance, whereas P4HA1 showed as specific to the CLCA cohort.

Epistatic interactions reveal pathway co‑operation between JAK1 and CTNNB1 (LRT P = 0.0066). Specifically, JAK1 L910P - CTNNB1 T41I (P = 0.008) and JAK1 S729C - CTNNB1 D32N (P = 0.015) are significant two pairs of mutant show epistatic relationships.

Survival analysis validates the significance of TP53 gene mutation (P = 0.018) in survival difference. However, for P4HA1, CTNNB1, and JAK1, the log-rank tests did not show statistical significance in terms of overall survival.

**Discussion**

Our primary objective was to move beyond mutation frequency and P‑value rankings to quantify the true selective impact of somatic variants in hepatocellular carcinoma (HCC). By applying the “cancereffectsizeR” framework to 494 CLCA genomes, we confirmed that CTNNB1, TP53, and JAK1 carry the largest scaled selection coefficients, underscoring their dominant evolutionary roles in HCC. Importantly, we uncovered P4HA1 D372A as a previously unrecognized but strongly selected variant. Although rare, its effect size rivals classical CTNNB1 hotspots, highlighting how effect‑size modelling can elevate low‑prevalence yet high‑impact events that frequency‑based screens overlook.

Comparative analysis with the American‑enriched TCGA‑LIHC cohort showed that most high‑effect variants of CTNNB1 and TP53 are shared across populations, indicating a conserved core driver spectrum. Nevertheless, China‑specific P4HA1 hit indicates potential population skew in selective landscapes, thereby informing geographically/racial tailored therapeutic or screening strategies. Moreover, recent research indicates that P4HA1 plays a significant role in cancer progression mechanism highlighting the importance of our study [8].

Epistasis testing revealed synergistic interactions between JAK1 and CTNNB1, both at the gene level (LRT P = 0.0066) and at specific variant pairs (JAK1 L910P - CTNNB1 T41I; JAK1 S729C - CTNNB1 D32N). Such positive epistasis suggests combined mutation may confers an amplified fitness advantage for cancer development, providing a new insight for HCC development.

TP53 mutations show a statistically significant in overall survival, aligning with their high selection coefficients. CTNNB1 and JAK1 deliver large fitness advantages to the tumor cell, but their net effect on patient survival is neutral here, underscoring the multifactorial nature of clinical outcomes.  P4HA1 illustrates that a potent but rare driver.

Overall, several limitations exist in our study. Focusing on a single, highly selected gene can be misleading because cancers evolve within complex, interactive networks. Although effect‑size ranks highlight potential drivers, mutations over those does not necessarily worsen patient survival. Once cancer is established, the outcome may depend on various factors. In survival analysis, we were not able to access the clinical data for CLCA cohort, disabled comparison of overall survivals. Even for the TCGA cohort, sample size is the main constraint, especially for JAK and P4HA1.

In conclusion, our selection‑effect estimation, cross‑cohort comparison, and epistasis analysis delivers a refined hierarchy of oncogenic events in HCC. This approach (i) prioritizes both common and rare high‑effect drivers for different cohorts, (ii) reveals population‑specific vulnerabilities, and (iii) identifies cooperative mutation pairs that could guide combination therapies and showed potential prognostic effect resulted by driver gene/variant mutations. Collectively, our results may bring new insight for advancing precision oncology for liver cancers.

**Reference:**

1. Sung H, Ferlay J, Siegel RL, Laversanne M, Soerjomataram I, Jemal A, et al. Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA Cancer J Clin.* 2021;71(3):209‑49.
2. Bagger FO, Borgwardt L, Jespersen AS, Brunak S, Lindgren C, Falgreen M, et al. Whole‑genome sequencing in clinical practice. *BMC Med Genomics.* 2024;17:39. doi:10.1186/s12920‑024‑01795‑w
3. Chen L, Zhang C, Xue R, Pei R, Wang G, Zhang S, et al. Deep whole‑genome analysis of 494 hepatocellular carcinomas. *Nature.* 2024;627:586‑93. doi:10.1038/s41586‑024‑07054‑3
4. Lawrence MS, Stojanov P, Mermel CH, Robinson JT, Garraway LA, Golub TR, et al. Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature.* 2014;505(7484):495‑501.
5. Mandell JD, Cannataro VL, Townsend JP. Estimation of neutral mutation rates and quantification of somatic variant selection using cancereffectsizeR. *Cancer Res.* 2023;83(4):500‑5. doi:10.1158/0008‑5472.CAN‑22‑1508
6. Tate JG, Bamford S, Jubb HC, Sondka Z, Beare DM, Bindal N, et al. COSMIC: the catalogue of somatic mutations in cancer. *Nucleic Acids Res.* 2019;47(D1):D941‑7.
7. Martincorena I, Raine KM, Gerstung M, Dawson KJ, Haase K, Van Loo P, et al. Universal patterns of selection in cancer and somatic tissues. *Cell.* 2017;171(5):1029‑41.e21. doi:10.1016/j.cell.2017.09.042
8. Yang X, Zhang D, Li M, Shao Y, Zhang X, Xue Y. P4HA1: an important target for treating fibrosis‑related diseases and cancer. *Front Pharmacol.* 2024;15:1493420. doi:10.3389/fphar.2024.1493420

**Appendix A:**

**A pie chart with different colored circles

AI-generated content may be incorrect.**

**Figure 1.** Genes and variants with strong selection coefficients.

A close-up of a chart

AI-generated content may be incorrect.

**Figure 2.** Epistasis heat map (P values) ofselected driver genes and variants.

A graph of a graph

AI-generated content may be incorrect.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gene Mutation (Yes/No)** | **TP53** | **CTNNB1** | **JAK1** | **P4HA1** |
| **Log-Rank Test P-value** | **0.018** | **0.898** | **0.912** | **0.542** |

**Figure 3.** K-M Curve for overall survival in TCGA-LIHC cohort with selected driver genes.

**Table 1.** Log-rank test results of selected driver genes in TCGA-LIHC cohort.

**Appendix B:**

Code 1 and 2 are R scripts that required to be processed under the same environment for cancereffectsizeR pipeline analysis. Code 3 is for survival analysis.

**Code 1:**

library(cancereffectsizeR)

library(data.table)

library(dplyr)

library(tidyr)

library(ggplot2)

library(scales)

library(VennDiagram)

library(grid)

library(pheatmap)

library(viridis)

# Somatic variant WGS data loading

wgs\_maf\_file <- read.table("C:/Yale/BIS 682/hcc\_clca\_2024/data\_mutations.txt", skip = 1, header = T, sep = "\t")

wgs\_maf <- preload\_maf(maf = wgs\_maf\_file, refset = "ces.refset.hg19")

wgs\_maf = wgs\_maf[germline\_variant\_site == F][repetitive\_region == F|cosmic\_site\_tier %in% 1:3]

# Clinical data

clinical\_data\_file <- read.table("C:/Yale/BIS 682/hcc\_clca\_2024/data\_clinical\_patient.txt",

skip = 1, header = T, sep = "\t")

setnames(clinical\_data\_file, "PATIENT\_ID", "Unique\_Patient\_Identifier")

head(clinical\_data\_file)

# Create cancereffectsizeR analysis and load data/whole genome sequencing

cesa <- CESAnalysis(refset = "ces.refset.hg19")

cesa <- load\_maf(cesa = cesa, maf = wgs\_maf, coverage = "genome",maf\_name = "test")

# Infer trinculeotide-context-specific relative rates of SNV mutation from

signature\_exclusions <- suggest\_cosmic\_signature\_exclusions(cancer\_type = "LIHC", treatment\_naive = TRUE)

cesa <- trinuc\_mutation\_rates(

cesa = cesa, signature\_set = ces.refset.hg19$signatures$COSMIC\_v3.2,

signature\_exclusions = signature\_exclusions

)

# Estimate neutral gene mutation rates using dNdScv, with tissue-specific mutation rate covariates.

cesa <- gene\_mutation\_rates(cesa, covariates = ces.refset.hg19$covariates$LIHC)

# neutral gene mutation rates extracted from dNdScv’s regression:

# View top dNdScv genes, sorted by significance

dndscv\_results2 <- cesa$dNdScv\_results[[1]]

sig\_genes2 <- dndscv\_results2[qallsubs\_cv < .05][order(qallsubs\_cv)][1:10]

sig\_genes2

# Infer scaled selection coefficients under the default model of clonal selection.

# By default, inference is restricted to recurrent mutations.

cesa <- ces\_variant(cesa = cesa, run\_name = "recurrents")

# Visualize top-effect variants.

p <- plot\_effects(

effects = cesa$selection$recurrents,

label\_individual\_variants = FALSE

)

# 2. add a scientific‐notation scale on the x axis

p +

scale\_x\_log10(

labels = label\_scientific(2)

)

plot\_effects(effects = cesa$selection$recurrents,

#group\_by = "gene",

label\_individual\_variants = F)

a = unique(cesa$selection$recurrents$variant\_name)

b = unique(cesaT$selection$example$variant\_name)

a\_clean <- gsub("\_", " ", unique(cesa$selection$recurrents$variant\_name))

b\_clean <- gsub("\_", "", unique(cesaT$selection$example$variant\_name))

a\_clean = unique(cesa$selection$recurrents$gene)

b\_clean = unique(cesaT$selection$example$gene)

# compute counts

nA <- length(a\_clean)

nB <- length(b\_clean)

nAB <- length(intersect(a\_clean, b\_clean))

# draw

draw.pairwise.venn(

area1 = nA,

area2 = nB,

cross.area = nAB,

category = c("recurrents", "example"),

fill = c("skyblue", "pink"),

lty = "blank",

cex = 2, # label size

cat.cex = 1.5, # category name size

cat.pos = c(-20, 20)

)

plot\_effects(cesa$selection$recurrents,

group\_by = "gene", topn = 8,

label\_individual\_variants = F)

# Attribute effects to mutational signatures

mut\_effects <- mutational\_signature\_effects(cesa, cesa$selection$recurrents)

# Plot a comparison of how signatures contribute to mutation vs. selection

plot\_signature\_effects(mut\_effects, viridis\_option = "F", num\_sig\_groups = 5)

# Structure of protein and match with the effect size

# epistasis analysis; mutually exclusive information?

# pdb protein structure file;

# slack channel: figures and draft response

genes <- c("CTNNB1", "TP53", "P4HA1", "JAK1", "NFIC", "PIK3CA")

# Get variants in the genes of interest that have sequencing coverage in all samples

variants <- select\_variants(cesa, genes = genes)

cesa <- ces\_gene\_epistasis(cesa = cesa, genes = genes, variants = variants, run\_name = "gene\_epistasis\_example1")

# 1) your pairwise gene‐epistasis result

# 2. get the full set of variants

vars <- sort(unique(c(df$variant\_A, df$variant\_B)))

# 3. init an empty square matrix

mat <- matrix(

NA\_real\_,

nrow = length(vars), ncol = length(vars),

dimnames = list(vars, vars)

)

# 4. fill the upper‐triangle with your A→B p-values

mat[cbind(df$variant\_A, df$variant\_B)] <- df$p\_epistasis

# 5. mirror to the lower triangle

mat[lower.tri(mat)] <- t(mat)[lower.tri(mat)]

# 6. blank out diagonal if you like

diag(mat) <- NA

# 7. plot

pheatmap(

mat,

color = viridis(100),

na\_col = "white"

)

# variant level epistasis

# Start by pulling full variant IDs (with protein identifier) from variants table

####

var\_list <- c(

"CTNNB1\_D32G","CTNNB1\_D32N","CTNNB1\_D32V","CTNNB1\_H36P",

"CTNNB1\_S33P","CTNNB1\_S37C","CTNNB1\_S37F","CTNNB1\_S45F",

"CTNNB1\_S45P","CTNNB1\_T41A","CTNNB1\_T41I",

"JAK1\_L910P","JAK1\_S729C",

"NFIC\_A393V",

"PIK3CA\_H1047R",

"TP53\_H193R","TP53\_R249S","TP53\_V157F",

"P4HA1\_D372A"

)

# 2. Generate all pairwise combinations of those variants

pair\_list <- combn(var\_list, 2, simplify = FALSE)

# 3. Lookup their variant\_id’s in your cesa$variants table

variant\_id\_pairs <- lapply(pair\_list, function(v) {

cesa$variants[v, variant\_id, on = "variant\_name"]

})

# 4. Run epistasis on every pair at once

cesa <- ces\_epistasis(

cesa = cesa,

variants = variant\_id\_pairs,

conf = .95,

run\_name = "all\_variant\_pairwise\_epistasisAll2"

)

# 1. pull out your variant‐level epistasis results

df <- cesa$epistasis$all\_variant\_pairwise\_epistasisAll2 %>%

# strip off the ENSP suffix right away

mutate(

variant\_A = gsub("(\_ENSP).\*", "", variant\_A),

variant\_B = gsub("(\_ENSP).\*", "", variant\_B)

)

# 2. get the full list of variants

vars <- sort(unique(c(df$variant\_A, df$variant\_B)))

# 3. initialize an empty square matrix

mat <- matrix(

NA\_real\_,

nrow = length(vars),

ncol = length(vars),

dimnames = list(vars, vars)

)

# 4. fill only the upper triangle from your data.frame

for (i in seq\_len(nrow(df))) {

a <- df$variant\_A[i]

b <- df$variant\_B[i]

mat[a, b] <- df$p\_epistasis[i]

}

# 5. mirror to lower triangle & blank diagonal

mat[lower.tri(mat)] <- t(mat)[lower.tri(mat)]

diag(mat) <- NA

# 6. plot

pheatmap(

mat,

color = viridis(100),

na\_col = "white",

angle\_col = 45

)

df%>%

filter(p\_epistasis < 0.05)

**Code 2:**

library(cancereffectsizeR)

library(data.table)

library(dplyr)

tcga\_maf\_file <- "TCGA-LIHC.maf.gz"

if (!file.exists(tcga\_maf\_file)) {

get\_TCGA\_project\_MAF(project = "LIHC", filename = tcga\_maf\_file)

}

# Prepare data

maf <- preload\_maf(maf = "cohortMAF.2025-04-23.maf.gz", refset = "ces.refset.hg38")

# Create cancereffectsizeR analysis and load data

cesaT <- CESAnalysis(refset = "ces.refset.hg38")

cesaT <- load\_maf(cesa = cesaT, maf = maf)

signature\_exclusions <- suggest\_cosmic\_signature\_exclusions(cancer\_type = "LIHC", treatment\_naive = TRUE)

cesaT <- trinuc\_mutation\_rates(

cesa = cesaT, signature\_set = ces.refset.hg38$signatures$COSMIC\_v3.4,

signature\_exclusions = signature\_exclusions

)

# Estimate neutral gene mutation rates using dNdScv, with tissue-specific mutation rate covariates.

cesaT <- gene\_mutation\_rates(cesaT, covariates = ces.refset.hg38$covariates$LIHC)

head(cesaT$gene\_rates)

dndscv\_results = cesaT$dNdScv\_results[[1]]

sig\_genes = dndscv\_results[qallsubs\_cv <= .05][order(qallsubs\_cv)]

sig\_genes

cesaT <- ces\_variant(cesaT, run\_name = "example")

# Visualize top-effect variants.

select\_gene = c("P4HA1","JAK1", "CTNNB1","ALB","TP53","FGA","HNF1A","PRDM11","CDKN1B","BMP5","ECHS1","AXIN1",

"ARID1A","TLE1")

plot\_effects(effects = cesaT$selection$example %>%

filter(gene %in% select\_gene),

#group\_by = "gene",

label\_individual\_variants = F)

# Attribute effects to mutational signatures

mut\_effects <- mutational\_signature\_effects(cesaT, cesaT$selection$example)

# Plot a comparison of how signatures contribute to mutation vs. selection

plot\_signature\_effects(mut\_effects, viridis\_option = "F")

**Code 3:**

#################################################################################

# overall survival status, overall survival (months)

# K-M curve and Log Rank Test

setwd("~/Desktop/BIS 681/Cancer Effect Size Project/Comparison\_TCGA\_survival")

# load the required packages

library(readr)

library(survival)

library(survminer)

library(dplyr)

library(survMisc)

# read in the data

all\_clinical <- read\_tsv("tcga\_all\_clinical.tsv", quote = "", col\_names = TRUE,

name\_repair = "minimal")

TP53\_mutation <- read\_tsv("TP53\_mutations.tsv", quote = "", col\_names = TRUE,

name\_repair = "minimal")

CTNNB1\_mutation <- read\_tsv("CTNNB1\_mutations.tsv", quote = "", col\_names = TRUE,

name\_repair = "minimal")

JAK1\_mutation <- read\_tsv("JAK1\_mutations.tsv", quote = "", col\_names = TRUE,

name\_repair = "minimal")

P4HA1\_mutation <- read\_tsv("P4HA1\_mutations.tsv", quote = "", col\_names = TRUE,

name\_repair = "minimal")

TLE1\_mutation <- read\_tsv("TLE1\_mutations.tsv", quote = "", col\_names = TRUE,

name\_repair = "minimal")

# extract all "Patient ID" lists from each of the gene mutation datasets

TP53\_mutation\_patients <- TP53\_mutation$`Patient ID`

CTNNB1\_mutation\_patients <- CTNNB1\_mutation$`Patient ID`

JAK1\_mutation\_patients <- JAK1\_mutation$`Patient ID`

P4HA1\_mutation\_patients <- P4HA1\_mutation$`Patient ID`

TLE1\_mutation\_patients <- TLE1\_mutation$`Patient ID`

# create a new mutation indicator column in the complete clinical dataset

# for each type of gene mutation

all\_clinical <- all\_clinical %>%

mutate(TP53\_mutation = ifelse(`Patient ID` %in% TP53\_mutation\_patients, "Yes", "No"),

CTNNB1\_mutation = ifelse(`Patient ID` %in% CTNNB1\_mutation\_patients,

"Yes", "No"),

JAK1\_mutation = ifelse(`Patient ID` %in% JAK1\_mutation\_patients, "Yes", "No"),

P4HA1\_mutation = ifelse(`Patient ID` %in% P4HA1\_mutation\_patients, "Yes", "No"),

TLE1\_mutation = ifelse(`Patient ID` %in% TLE1\_mutation\_patients, "Yes", "No"))

# create a new column for the overall survival status

all\_clinical <- all\_clinical %>%

mutate(OS\_status = ifelse(`Overall Survival Status` == "1:DECEASED", 1, 0))

# create a new column for the overall survival time in months

all\_clinical <- all\_clinical %>%

mutate(OS\_time = `Overall Survival (Months)`)

# plot the K-M curve and conduct the log rank test for each type of gene mutation

# TP53 mutation

km\_fit\_TP53 <- survfit(Surv(OS\_time, OS\_status) ~ TP53\_mutation, data = all\_clinical)

ggsurvplot(km\_fit\_TP53, data = all\_clinical, pval = TRUE, conf.int = TRUE,

title = "K-M Curve for TP53 Mutation",

xlab = "Overall Survival Time (Months)",

ylab = "Survival Probability")

log\_rank\_TP53 <- survdiff(Surv(OS\_time, OS\_status) ~ TP53\_mutation, data = all\_clinical)

log\_rank\_TP53\_pvalue <- 1 - pchisq(log\_rank\_TP53$chisq,

df = length(log\_rank\_TP53$n) - 1)

# CTNNB1 mutation

km\_fit\_CTN <- survfit(Surv(OS\_time, OS\_status) ~ CTNNB1\_mutation, data = all\_clinical)

ggsurvplot(km\_fit\_CTN, data = all\_clinical, pval = TRUE, conf.int = TRUE,

title = "K-M Curve for CTNNB1 Mutation",

xlab = "Overall Survival Time (Months)",

ylab = "Survival Probability")

log\_rank\_CTN <- survdiff(Surv(OS\_time, OS\_status) ~ CTNNB1\_mutation, data = all\_clinical)

log\_rank\_CTN\_pvalue <- 1 - pchisq(log\_rank\_CTN$chisq, df = length(log\_rank\_CTN$n) - 1)

# JAK1 mutation

km\_fit\_JAK1 <- survfit(Surv(OS\_time, OS\_status) ~ JAK1\_mutation, data = all\_clinical)

ggsurvplot(km\_fit\_JAK1, data = all\_clinical, pval = TRUE, conf.int = TRUE,

title = "K-M Curve for JAK1 Mutation",

xlab = "Overall Survival Time (Months)",

ylab = "Survival Probability")

log\_rank\_JAK1 <- survdiff(Surv(OS\_time, OS\_status) ~ JAK1\_mutation, data = all\_clinical)

log\_rank\_JAK1\_pvalue <- 1 - pchisq(log\_rank\_JAK1$chisq, df = length(log\_rank\_JAK1$n) - 1)

# P4HA1 mutation

km\_fit\_P4HA1 <- survfit(Surv(OS\_time, OS\_status) ~ P4HA1\_mutation, data = all\_clinical)

ggsurvplot(km\_fit\_P4HA1, data = all\_clinical, pval = TRUE, conf.int = TRUE,

title = "K-M Curve for P4HA1 Mutation",

xlab = "Overall Survival Time (Months)",

ylab = "Survival Probability")

log\_rank\_P4HA1 <- survdiff(Surv(OS\_time, OS\_status) ~ P4HA1\_mutation,

data = all\_clinical)

log\_rank\_P4HA1\_pvalue <- 1 - pchisq(log\_rank\_P4HA1$chisq,

df = length(log\_rank\_P4HA1$n) - 1)

# TLE1 mutation

km\_fit\_TLE1 <- survfit(Surv(OS\_time, OS\_status) ~ TLE1\_mutation, data = all\_clinical)

ggsurvplot(km\_fit\_TLE1, data = all\_clinical, pval = TRUE, conf.int = TRUE,

title = "K-M Curve for TLE1 Mutation",

xlab = "Overall Survival Time (Months)",

ylab = "Survival Probability")

log\_rank\_TLE1 <- survdiff(Surv(OS\_time, OS\_status) ~ TLE1\_mutation, data = all\_clinical)

log\_rank\_TLE1\_pvalue <- 1 - pchisq(log\_rank\_TLE1$chisq, df = length(log\_rank\_TLE1$n) - 1)

# create a summary table for the log rank test results

log\_rank\_results <- data.frame(

Gene\_Mutation = c("TP53", "CTNNB1", "JAK1", "P4HA1", "TLE1"),

p\_value = c(log\_rank\_TP53\_pvalue, log\_rank\_CTN\_pvalue, log\_rank\_JAK1\_pvalue,

log\_rank\_P4HA1\_pvalue, log\_rank\_TLE1\_pvalue)

)

# print the summary table

print(log\_rank\_results)

#################################################################################

# Disease Free Status, Disease Free (Months)

# K-M curve and Log Rank Test

# create a new column for the disease free status

all\_clinical <- all\_clinical %>%

mutate(DFS\_status = ifelse(`Disease Free Status` == "1:Recurred/Progressed", 1, 0))

# create a new column for the disease free time in months

all\_clinical <- all\_clinical %>%

mutate(DFS\_time = `Disease Free (Months)`)

# plot the K-M curve and conduct the log rank test for each type of gene mutation

# TP53 mutation

km\_fit\_TP53 <- survfit(Surv(DFS\_time, DFS\_status) ~ TP53\_mutation, data = all\_clinical)

ggsurvplot(km\_fit\_TP53, data = all\_clinical, pval = TRUE, conf.int = TRUE,

title = "K-M Curve for TP53 Mutation",

xlab = "Disease Free Time (Months)",

ylab = "Survival Probability")

log\_rank\_TP53 <- survdiff(Surv(DFS\_time, DFS\_status) ~ TP53\_mutation,

data = all\_clinical)

log\_rank\_TP53\_pvalue <- 1 - pchisq(log\_rank\_TP53$chisq,

df = length(log\_rank\_TP53$n) - 1)

# CTNNB1 mutation

km\_fit\_CTN <- survfit(Surv(DFS\_time, DFS\_status) ~ CTNNB1\_mutation, data = all\_clinical)

ggsurvplot(km\_fit\_CTN, data = all\_clinical, pval = TRUE, conf.int = TRUE,

title = "K-M Curve for CTNNB1 Mutation",

xlab = "Disease Free Time (Months)",

ylab = "Survival Probability")

log\_rank\_CTN <- survdiff(Surv(DFS\_time, DFS\_status) ~ CTNNB1\_mutation,

data = all\_clinical)

log\_rank\_CTN\_pvalue <- 1 - pchisq(log\_rank\_CTN$chisq,

df = length(log\_rank\_CTN$n) - 1)

# JAK1 mutation

km\_fit\_JAK1 <- survfit(Surv(DFS\_time, DFS\_status) ~ JAK1\_mutation, data = all\_clinical)

ggsurvplot(km\_fit\_JAK1, data = all\_clinical, pval = TRUE, conf.int = TRUE,

title = "K-M Curve for JAK1 Mutation",

xlab = "Disease Free Time (Months)",

ylab = "Survival Probability")

log\_rank\_JAK1 <- survdiff(Surv(DFS\_time, DFS\_status) ~ JAK1\_mutation,

data = all\_clinical)

log\_rank\_JAK1\_pvalue <- 1 - pchisq(log\_rank\_JAK1$chisq,

df = length(log\_rank\_JAK1$n) - 1)

# P4HA1 mutation

km\_fit\_P4HA1 <- survfit(Surv(DFS\_time, DFS\_status) ~ P4HA1\_mutation,

data = all\_clinical)

ggsurvplot(km\_fit\_P4HA1, data = all\_clinical, pval = TRUE, conf.int = TRUE,

title = "K-M Curve for P4HA1 Mutation",

xlab = "Disease Free Time (Months)",

ylab = "Survival Probability")

log\_rank\_P4HA1 <- survdiff(Surv(DFS\_time, DFS\_status) ~ P4HA1\_mutation,

data = all\_clinical)

log\_rank\_P4HA1\_pvalue <- 1 - pchisq(log\_rank\_P4HA1$chisq,

df = length(log\_rank\_P4HA1$n) - 1)

# TLE1 mutation

km\_fit\_TLE1 <- survfit(Surv(DFS\_time, DFS\_status) ~ TLE1\_mutation, data = all\_clinical)

ggsurvplot(km\_fit\_TLE1, data = all\_clinical, pval = TRUE, conf.int = TRUE,

title = "K-M Curve for TLE1 Mutation",

xlab = "Disease Free Time (Months)",

ylab = "Survival Probability")

log\_rank\_TLE1 <- survdiff(Surv(DFS\_time, DFS\_status) ~ TLE1\_mutation,

data = all\_clinical)

log\_rank\_TLE1\_pvalue <- 1 - pchisq(log\_rank\_TLE1$chisq,

df = length(log\_rank\_TLE1$n) - 1)

# create a summary table for the log rank test results

log\_rank\_results <- data.frame(

Gene\_Mutation = c("TP53", "CTNNB1", "JAK1", "P4HA1", "TLE1"),

p\_value = c(log\_rank\_TP53\_pvalue, log\_rank\_CTN\_pvalue, log\_rank\_JAK1\_pvalue,

log\_rank\_P4HA1\_pvalue, log\_rank\_TLE1\_pvalue))

# print the summary table

print(log\_rank\_results)