Survival Analysis with HGAT group and telomerase status

Run Jin

2/4/2022

This notebook will do the following survival analysis

1. Univariate analysis

- DMG H3K28 vs rest
- Telhunt score (separate into categories by 1.07)
- HGAT vs. non-HGAT for all samples
- ATRX (mut N/Y)
- ALT vs. non-ALT for all samples

2. Multivariate analysis

- DMG H3K28 vs rest + HGAT vs. non-HGAT + Telhunt score (categorical) + sex + ATRX (mut N/Y)
- DMG H3K28 vs rest + HGAT vs. non-HGAT + Telhunt score (continuous) + sex + ATRX (mut N/Y)
- DMG H3K28 vs rest + HGAT vs. non-HGAT + ALT vs. non-ALT + H3K28*ALT + Telhunt score (categorical) + H3K28+Telhunt + sex + ATRX (mut N/Y)
- DMG H3K28 vs rest + HGAT vs. non-HGAT + ALT vs. non-ALT + H3K28*ALT + Telhunt score (continuous) + H3K28+Telhunt + sex + ATRX (mut N/Y)

Packages and functions Read in set up script.

```
library(survival)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(ggpubr)
```

Set up directories

```r
root_dir <- rprojroot::find_root(rprojroot::has_dir(".git"))
analysis_dir <- file.path(root_dir, "analyses", "06-survival-analysis")

plots_dir <- file.path(analysis_dir, "plots")
if (!dir.exists(plots_dir)) {
  dir.create(plots_dir)
}

output_dir <- file.path(analysis_dir, "output")
if (!dir.exists(output_dir)) {
  dir.create(plots_dir)
}
```

Read in files

```r
# get the meta information
meta <- readr::read_tsv(file.path(root_dir,
                           "analyses/02-add-histologies/output/stundon_hgat_03312022_updated_hist_alt.ts
  # remove existing ones to get newer data
  dplyr::select(-c("OS_days", "OS_status"))
```

```
## Rows: 87 Columns: 115

## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (54): Kids_First_Biospecimen_ID_DNA, Kids_First_Biospecimen_ID_RNA, Kids...
## dbl (52): TH T/TH N, UBTF Binary, ATRX Reverse Binary, ATRX IHC Binary, ATRX...
## lgl  (9): cell_line_composition, DAXX_fusion, ...34, NF...35, call...37, CPG...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# get survival information
survival_v21 <- readr::read_tsv(file.path(root_dir,
                                   "analyses/02-add-histologies/input-v21/pbta-histologies.tsv")) %>%
  dplyr::select("Kids_First_Participant_ID", "OS_days", "OS_status") %>%
  distinct()
```

```
## Rows: 2840 Columns: 38

## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (33): Kids_First_Biospecimen_ID, sample_id, aliquot_id, Kids_First_Parti...
## dbl  (5): OS_days, age_last_update_days, normal_fraction, tumor_fraction, tu...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Organize data

```r
# join with meta
meta <- meta %>%
  dplyr::left_join(survival_v21) %>%
  dplyr::distinct(Kids_First_Participant_ID, .keep_all = TRUE)
```

```
## Joining, by = "Kids_First_Participant_ID"
```

```r
# recode for analysis
meta_formatted <- meta %>%
  # recode the categories -
  # DECEASED maps to a survival event status of 1, LIVING maps to a censored observation with value 0
  dplyr::mutate(OS_status_recoded = case_when(
    OS_status == "LIVING" ~ 0,
    OS_status == "DECEASED" ~1
 )) %>%
  # retain only ones with OS days
  dplyr::filter(!is.na(OS_days)) %>%
  # calculte the years
  dplyr::mutate(OS_years = OS_days / 365.25) %>%
  # categorize by telhunt scores
  dplyr::mutate(telhunt_cat = case_when(
    `TH T/TH N` > 1.07 ~ "High",
    `TH T/TH N` < 1.07 ~ "Low"
  )) %>%
  # categorize by DMG, H3K28 or not
  dplyr::mutate(dmg_h3k28 = case_when(
    grepl("DMG, H3 K28", molecular_subtype) ~ "DMG",
    TRUE ~ "non-DMG"
  )) %>%
  # categorize ATRX
  dplyr::mutate(atrx_mut = case_when(
    !is.na(`ATRX Mutation`) ~ "ATRX_mut",
    TRUE ~ "non_ATRX_mut"
  )) %>%
  # rename telhunt score and CCA
  dplyr::mutate(telhunt_score = `TH T/TH N`,
                cca_telhunt = `CCA Sept 2021`,
                alt_final = `alt final`)

# define as factor
meta_formatted$dmg_h3k28 <- factor(meta_formatted$dmg_h3k28, levels = c("non-DMG", "DMG"))
meta_formatted$telhunt_cat <- factor(meta_formatted$telhunt_cat, levels = c("Low", "High"))
meta_formatted$group <- factor(meta_formatted$group, levels = c("non-HGAT", "HGAT"))
meta_formatted$alt_final <- factor(meta_formatted$alt_final, levels = c("NEG", "POS"))
meta_formatted$atrx_mut <- factor(meta_formatted$atrx_mut, levels = c("non_ATRX_mut", "ATRX_mut"))

# define as numberic
meta_formatted$telhunt_score <- as.numeric(meta_formatted$telhunt_score)
meta_formatted_hgat <- meta_formatted %>%
  dplyr::filter(group == "HGAT")
```

A Run for all samples

Log Rank analysis

Generate output for categorical files

```
# for(ind_var in c("dmg_h3k28", "telhunt_cat", "group", "atrx_mut", "alt_final")){
#   # define model
#   model <- paste0("survival::Surv(time = OS_years, event = OS_status_recoded) ~ ", ind_var)
#
#   # run survival analysis
#   fit <- survival::survdiff(formula(model),
#                             data = meta_formatted)
#   # Obtain p value for Chi-Squared stat
#   fit$p.value <- pchisq(fit$chisq, df = length(fit$n) - 1, lower = FALSE)
#
#   # save the output
#   saveRDS(fit, file.path(output_dir, paste0("log_rank_survival_per_", ind_var, ".RDS")))
#
#   # generate plots fit
#   fit_plot <- survfit(formula(model), data = data_used)
#
#   # output the plot
#   plot_logrank <- survminer::ggsurvplot(fit_plot,
#                                         data=data_used,
#                                         xlim = c(0, 14),
#                                         break.time.by = 1,
#                                         pval = TRUE,
#                                         conf.int = TRUE,
#                                         risk.table = TRUE, # Add risk table
#                                         linetype = "strata", # Change line type by groups
#                                         surv.median.line = "hv", # Specify median survival
#                                         ggtheme = theme_bw())
#
#   print(plot_logrank)
#   # Make this plot a combined plot
#   surv_plot_logrank <- cowplot::plot_grid(plot_logrank[[1]],
#                                          plot_logrank[[2]],
#                                          nrow = 2,
#                                          rel_heights = c(2.5, 1))
#   # Save the plot
#   cowplot::save_plot(filename = file.path(plots_dir,
#                                          paste0("logrank_survival_by_", ind_var, ".png")),
#                      plot = surv_plot_logrank)
#
# }
```

Multivariate analysis

Multivariate analysis - DMG H3K28 vs rest + HGAT vs. non-HGAT + ALT vs. non-ALT+ Telhunt score (categorical) + sex + ATRX (mut N/Y) - DMG H3K28 vs rest + HGAT vs. non-HGAT + ALT vs. non-ALT

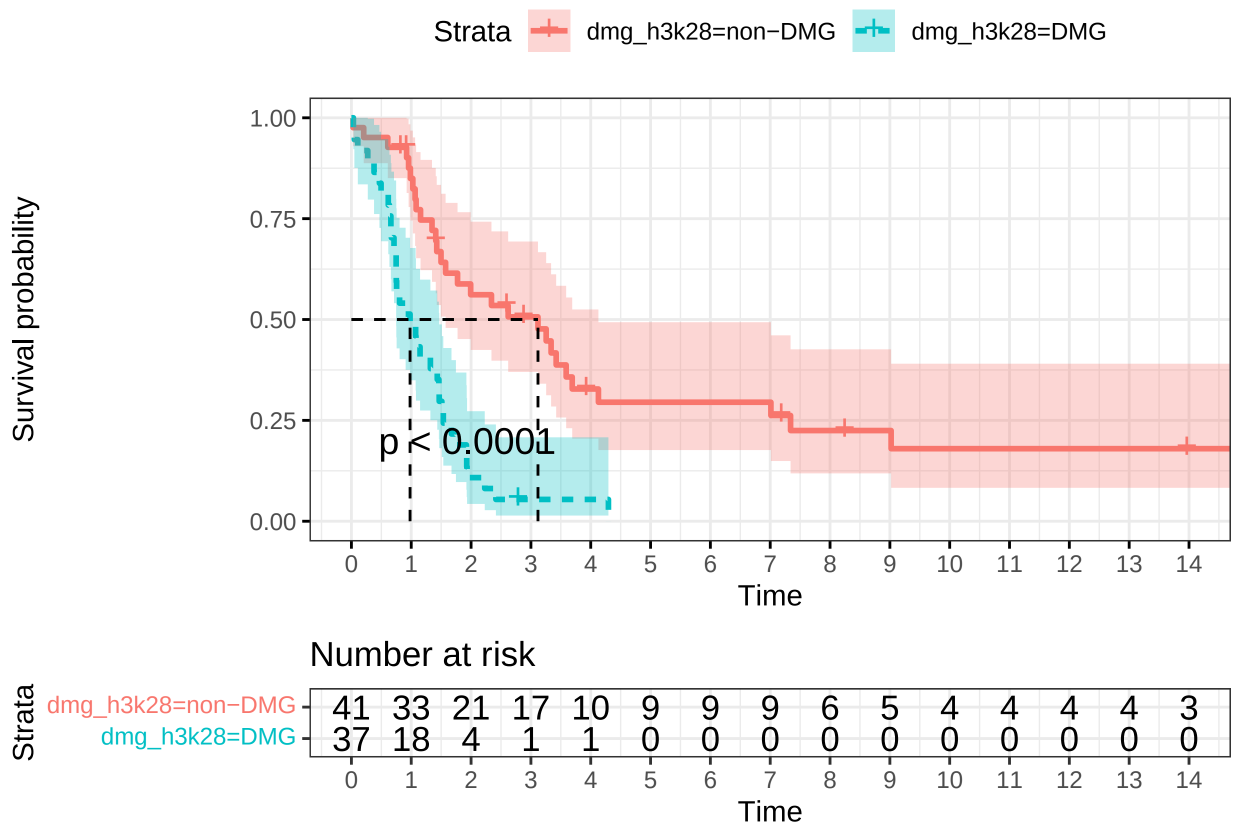
+ Telhunt score (continuous) + sex + ATRX (mut N/Y) - DMG H3K28 vs rest + HGAT vs. non-HGAT + ALT vs. non-ALT + H3K28*ALT + Telhunt score (categorical) + H3K28+Telhunt + sex + ATRX (mut N/Y) - DMG H3K28 vs rest + HGAT vs. non-HGAT + ALT vs. non-ALT + H3K28*ALT + Telhunt score (continuous) + H3K28+Telhunt + sex + ATRX (mut N/Y)

```r
# # define multi-variates that we are using for analyzing survival
# list_of_variates <- c("dmg_h3k28 + group + alt_final + telhunt_cat + germline_sex_estimate + atrx_mut
#                       "dmg_h3k28 + group + alt_final + telhunt_score + germline_sex_estimate + atrx_m
#                       "dmg_h3k28 + group + alt_final + telhunt_cat + germline_sex_estimate + atrx_mut
#                       "dmg_h3k28 + group + alt_final + telhunt_score + germline_sex_estimate + atrx_m
#                       )
#
# # define model
# for (ind_var in list_of_variates){
#   model <- paste0("survival::Surv(time = OS_years, event = OS_status_recoded) ~ ", ind_var)
#
#   # depending on which variables are used, data used will be different
#   data_used <- meta_formatted
#
#   fit <- survival::coxph(
#         formula(model),
#         data = data_used
#       )
#   # generate output
#   table <- broom::tidy(fit)
#
#   # Save the table data in a TSV
#   readr::write_tsv(table, file.path(output_dir, paste0("cox_reg_results_per_", ind_var, ".tsv")))
#
#   print(table)
#
#   # printout the plot
#   forest_coxph <- survminer::ggforest(fit, data = data_used)
#   print(forest_coxph)
#
# }
```

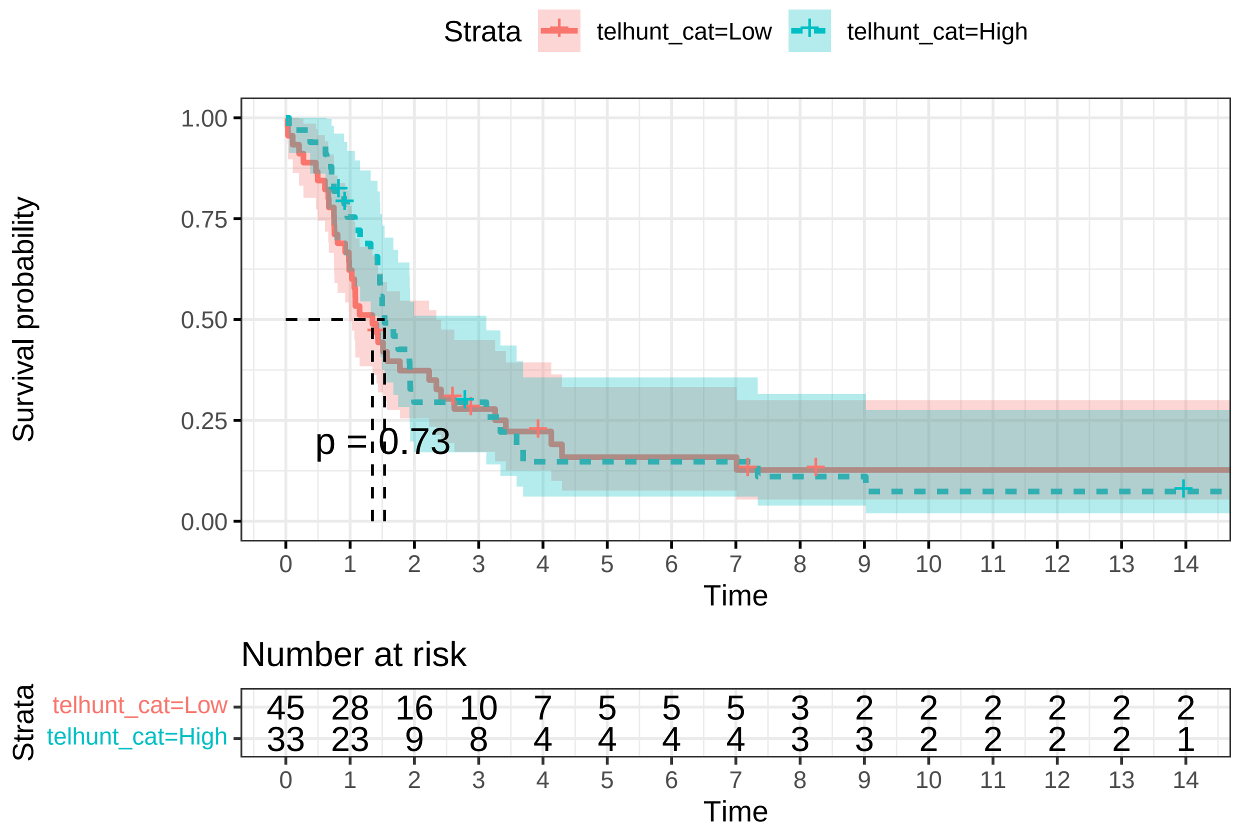
B run for HGAT only

Univariate analysis

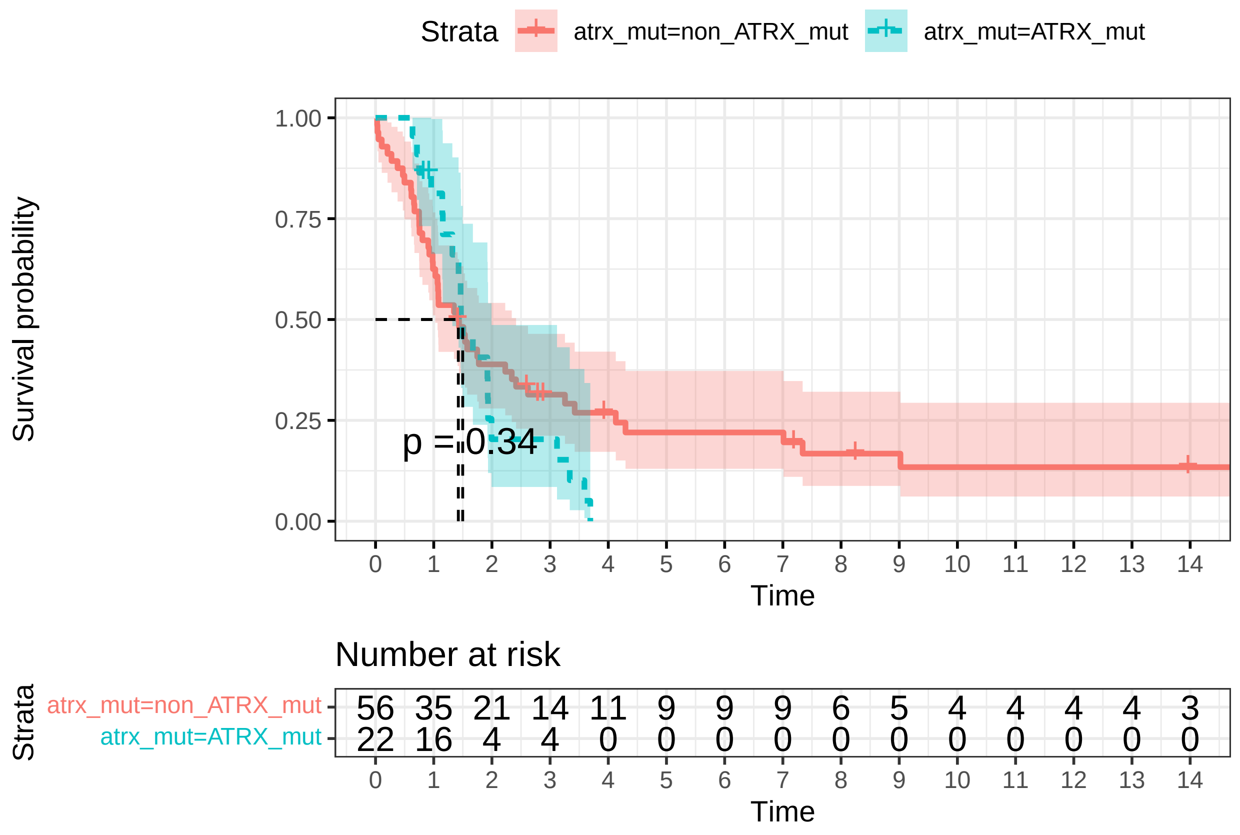
```r
for(ind_var in c("dmg_h3k28", "telhunt_cat", "atrx_mut", "alt_final")){
  # define model
  model <- paste0("survival::Surv(time = OS_years, event = OS_status_recoded) ~ ", ind_var)

  # depending on which variables are used, data used will be different
  data_used <- meta_formatted_hgat

  # run survival analysis
  fit <- survival::survdiff(formula(model),
                            data = data_used)
  # Obtain p value for Chi-Squared stat
  fit$p.value <- pchisq(fit$chisq, df = length(fit$n) - 1, lower = FALSE)

  # save the output
  saveRDS(fit, file.path(output_dir, paste0("log_rank_survival_per_", ind_var, "_os_only_hgat.RDS")))

  # generate plots fit
  fit_plot <- survfit(formula(model), data = data_used)

  # output the plot
  plot_logrank <- survminer::ggsurvplot(fit_plot,
                                        data=data_used,
                                        xlim = c(0, 14),
                                        break.time.by = 1,
                                        pval = TRUE,
                                        conf.int = TRUE,
                                        risk.table = TRUE, # Add risk table
                                        linetype = "strata", # Change line type by groups
                                        surv.median.line = "hv", # Specify median survival
                                        ggtheme = theme_bw())

  print(plot_logrank)
  # Make this plot a combined plot
  surv_plot_logrank <- cowplot::plot_grid(plot_logrank[[1]],
                                          plot_logrank[[2]],
                                          nrow = 2,
                                          rel_heights = c(2.5, 1))
  # Save the plot
  cowplot::save_plot(filename = file.path(plots_dir,
                                          paste0("logrank_survival_by_", ind_var, "_os_only_hgat.png"))
                     plot = surv_plot_logrank)


}
```



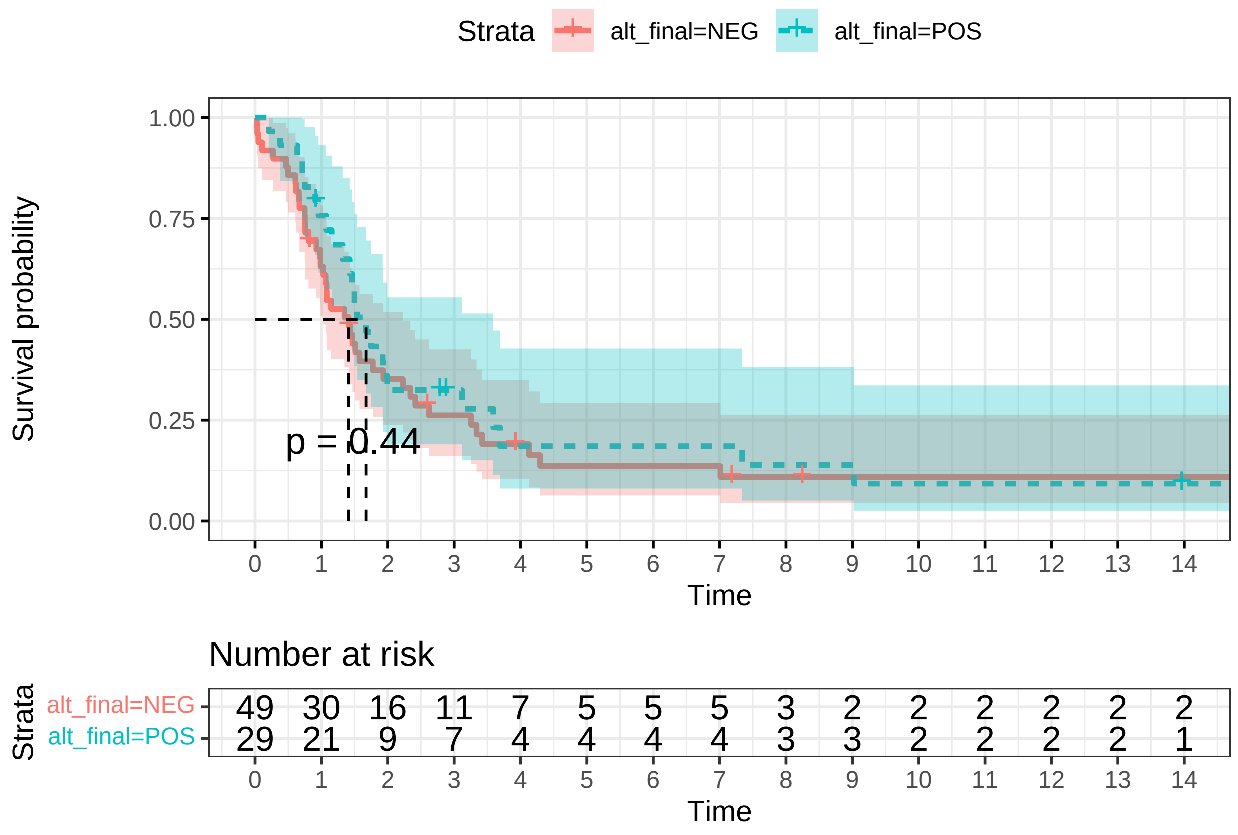
Strata	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
dmg_h3k28=non−DMG	41	33	21	17	10	9	9	9	6	5	4	4	4	4	3
dmg_h3k28=DMG	37	18	4	1	1	0	0	0	0	0	0	0	0	0	0



Strata	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
telhunt_cat=Low	45	28	16	10	7	5	5	5	3	2	2	2	2	2	2
telhunt_cat=High	33	23	9	8	4	4	4	4	3	3	2	2	2	2	1



Strata	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
atrx_mut=non_ATRX_mut	56	35	21	14	11	9	9	9	6	5	4	4	4	4	3
atrx_mut=ATRX_mut	22	16	4	4	0	0	0	0	0	0	0	0	0	0	0



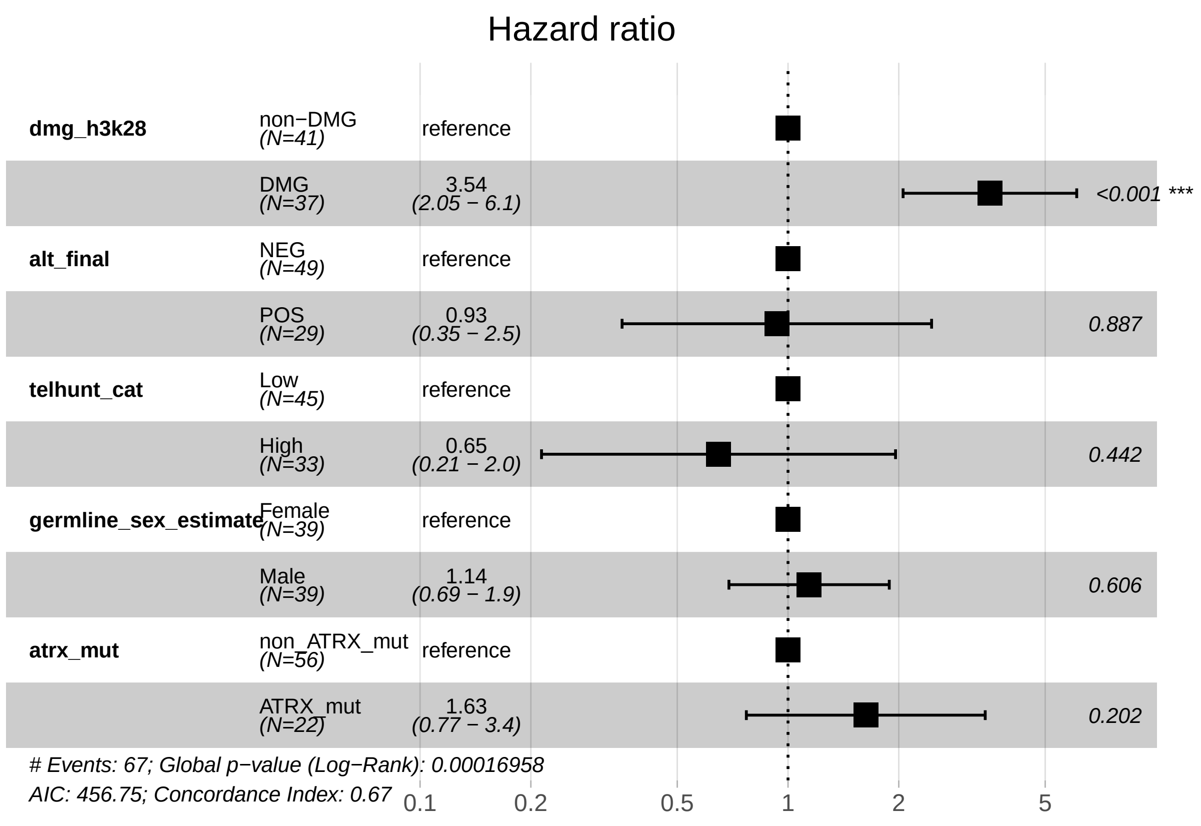
Multivariate analysis

Multivariate analysis for HGAT samples only - DMG H3K28 vs rest + ALT vs. non-ALT+ Telhunt score (categorical) + sex + ATRX (mut N/Y) - DMG H3K28 vs rest + ALT vs. non-ALT + Telhunt score (continuous) + sex + ATRX (mut N/Y) - DMG H3K28 vs rest + ALT vs. non-ALT + H3K28*ALT + Telhunt score (categorical) + H3K28+Telhunt + sex + ATRX (mut N/Y) - DMG H3K28 vs rest + ALT vs. non-ALT + H3K28*ALT + Telhunt score (continuous) + H3K28+Telhunt + sex + ATRX (mut N/Y)

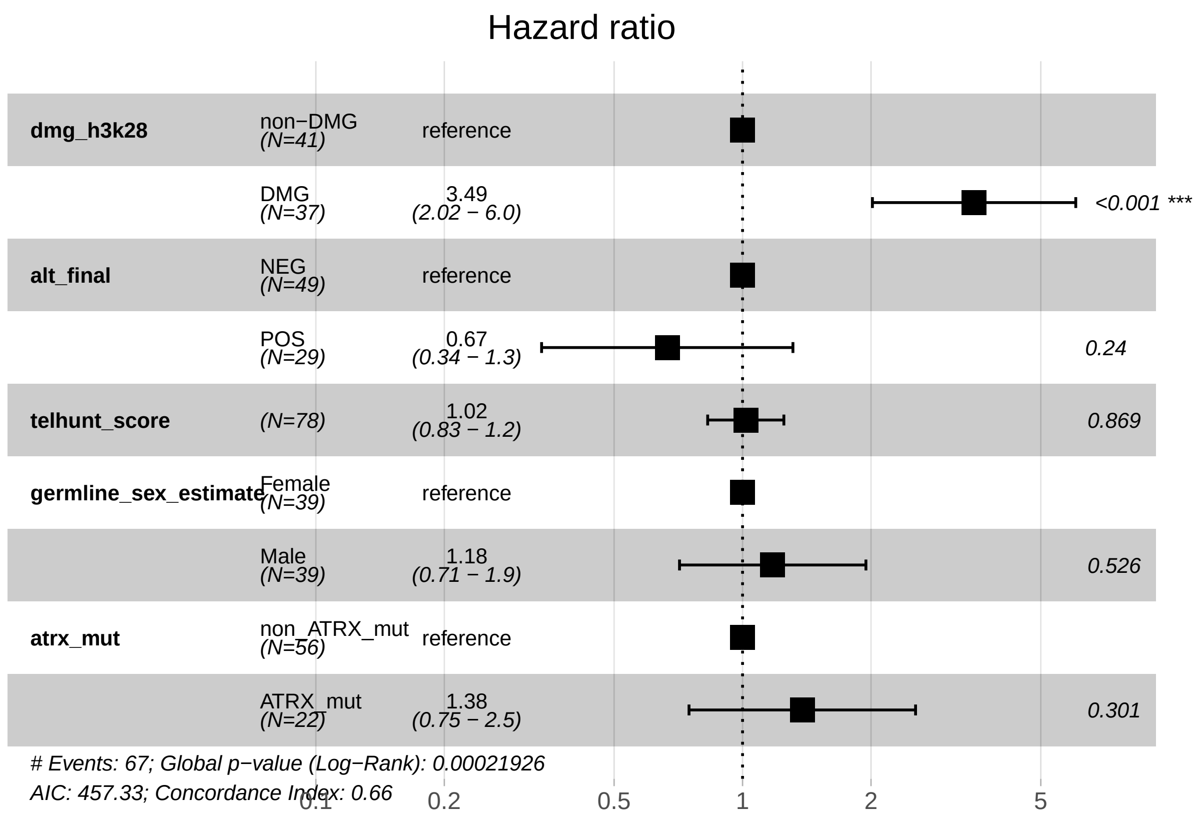
```
# define multi-variates that we are using for analyzing survival
list_of_variates <- c("dmg_h3k28 + alt_final + telhunt_cat + germline_sex_estimate + atrx_mut",
                      "dmg_h3k28 + alt_final + telhunt_score + germline_sex_estimate + atrx_mut",
                      "dmg_h3k28 + alt_final + telhunt_cat + germline_sex_estimate + atrx_mut + dmg_h3k
                      "dmg_h3k28 + alt_final + telhunt_score + germline_sex_estimate + atrx_mut + dmg_h3
                       )

# define model
for (ind_var in list_of_variates){
  model <- paste0("survival::Surv(time = OS_years, event = OS_status_recoded) ~ ", ind_var)

  # depending on which variables are used, data used will be different
  data_used <- meta_formatted_hgat

  fit <- survival::coxph(
        formula(model),
        data = data_used
```

```
    )
  # generate output
  table <- broom::tidy(fit)

  # Save the table data in a TSV
  readr::write_tsv(table, file.path(output_dir, paste0("cox_reg_results_per_", ind_var, "_os_only_hgat.

  print(table)

  # printout the plot
  forest_coxph <- survminer::ggforest(fit, data = data_used)
  print(forest_coxph)

}
```

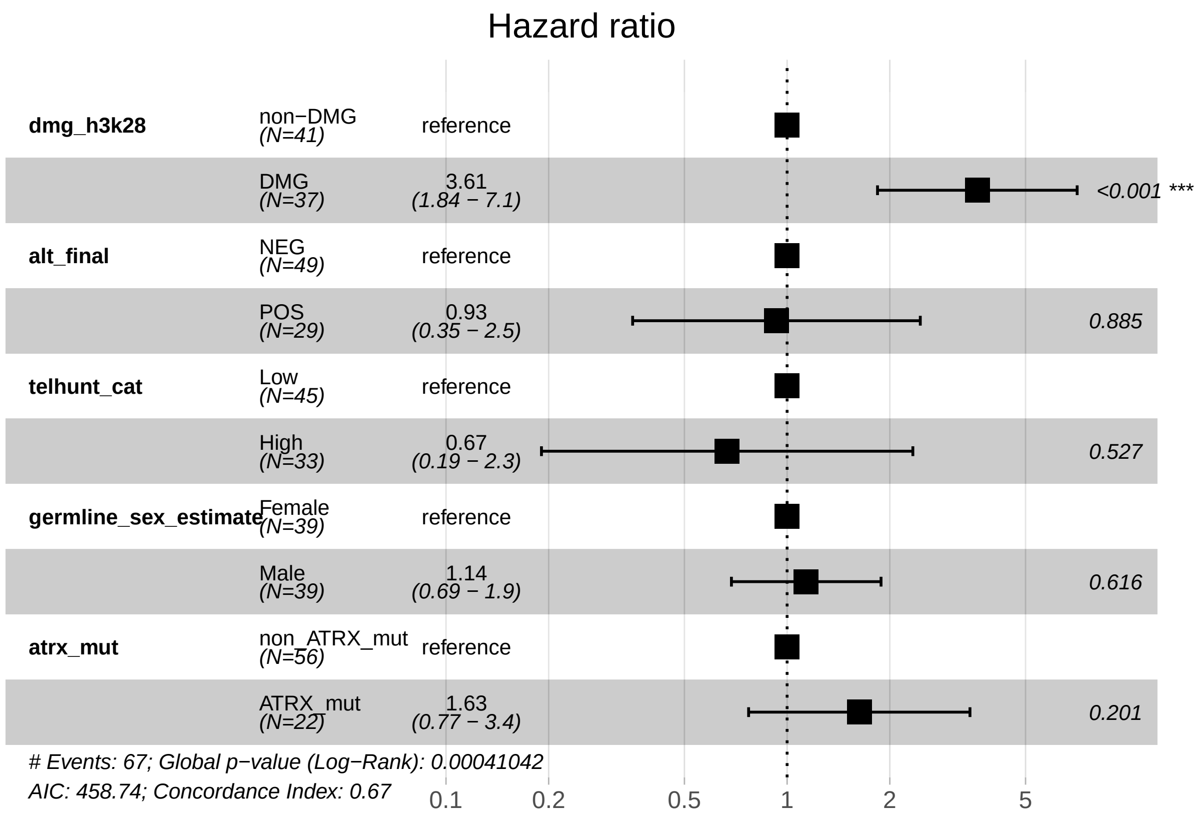
```
## # A tibble: 5 x 5
##   term                       estimate std.error statistic    p.value
##   <chr>                         <dbl>     <dbl>     <dbl>      <dbl>
## 1 dmg_h3k28DMG                 1.26       0.277     4.56  0.00000516
## 2 alt_finalPOS                -0.0699     0.494    -0.142 0.887
## 3 telhunt_catHigh             -0.435      0.565    -0.769 0.442
## 4 germline_sex_estimateMale    0.132      0.256     0.516 0.606
## 5 atrx_mutATRX_mut             0.487      0.381     1.28  0.202
```



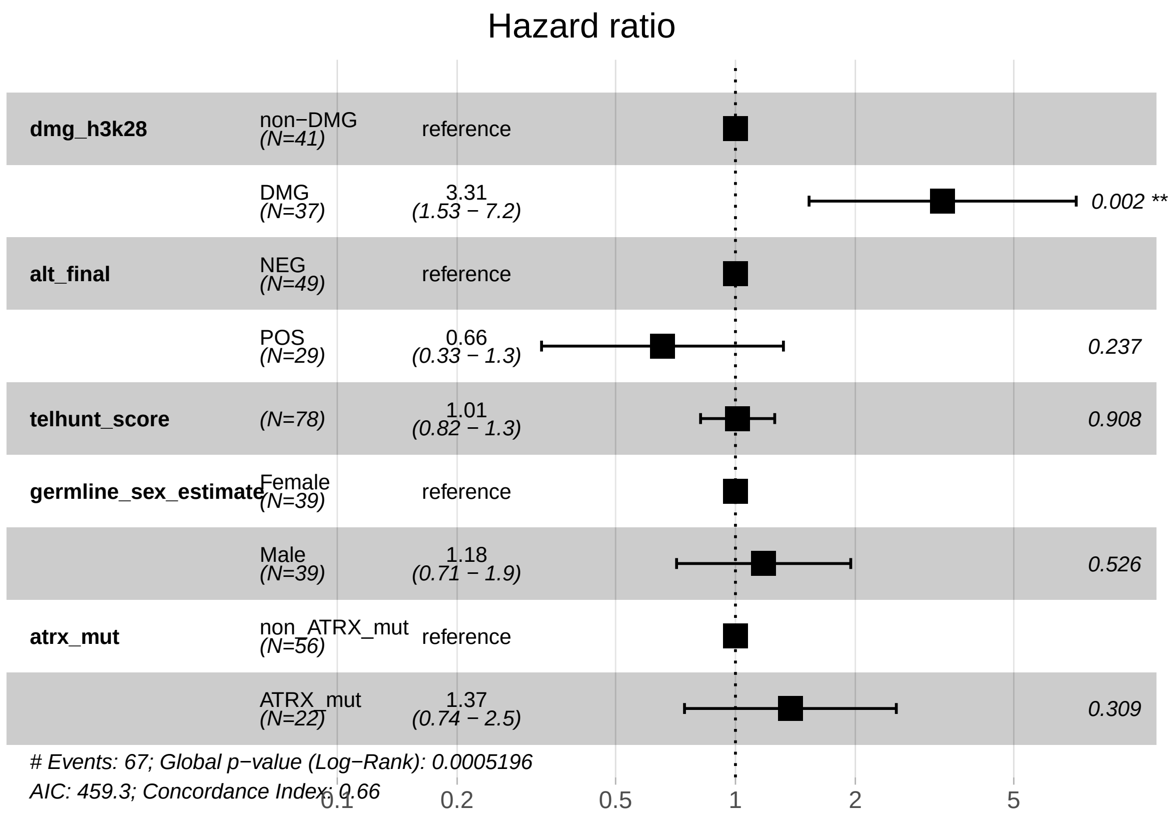
```
## # A tibble: 5 x 5
##   term                        estimate std.error statistic    p.value
##   <chr>                          <dbl>     <dbl>     <dbl>      <dbl>
## 1 dmg_h3k28DMG                  1.25       0.280     4.46  0.00000806
## 2 alt_finalPOS                 -0.406      0.346    -1.17  0.240
## 3 telhunt_score                 0.0173     0.105     0.165 0.869
## 4 germline_sex_estimateMale     0.163      0.257     0.634 0.526
## 5 atrx_mutATRX_mut              0.322      0.312     1.03  0.301
```



```
## # A tibble: 6 x 5
##   term                           estimate std.error statistic  p.value
##   <chr>                             <dbl>     <dbl>     <dbl>    <dbl>
## 1 dmg_h3k28DMG                     1.28       0.344     3.73  0.000188
## 2 alt_finalPOS                    -0.0719     0.495    -0.145 0.885
## 3 telhunt_catHigh                 -0.405      0.640    -0.633 0.527
## 4 germline_sex_estimateMale        0.129      0.258     0.501 0.616
## 5 atrx_mutATRX_mut                 0.487      0.381     1.28  0.201
## 6 dmg_h3k28DMG:telhunt_catHigh    -0.0520     0.515    -0.101 0.920
```



```
## # A tibble: 6 x 5
##   term                        estimate std.error statistic p.value
##   <chr>                          <dbl>     <dbl>     <dbl>   <dbl>
## 1 dmg_h3k28DMG                  1.20       0.394     3.04  0.00237
## 2 alt_finalPOS                 -0.422      0.357    -1.18  0.237
## 3 telhunt_score                 0.0127     0.109     0.116 0.908
## 4 germline_sex_estimateMale     0.163      0.257     0.634 0.526
## 5 atrx_mutATRX_mut              0.318      0.312     1.02  0.309
## 6 dmg_h3k28DMG:telhunt_score    0.0421     0.226     0.186 0.852
```



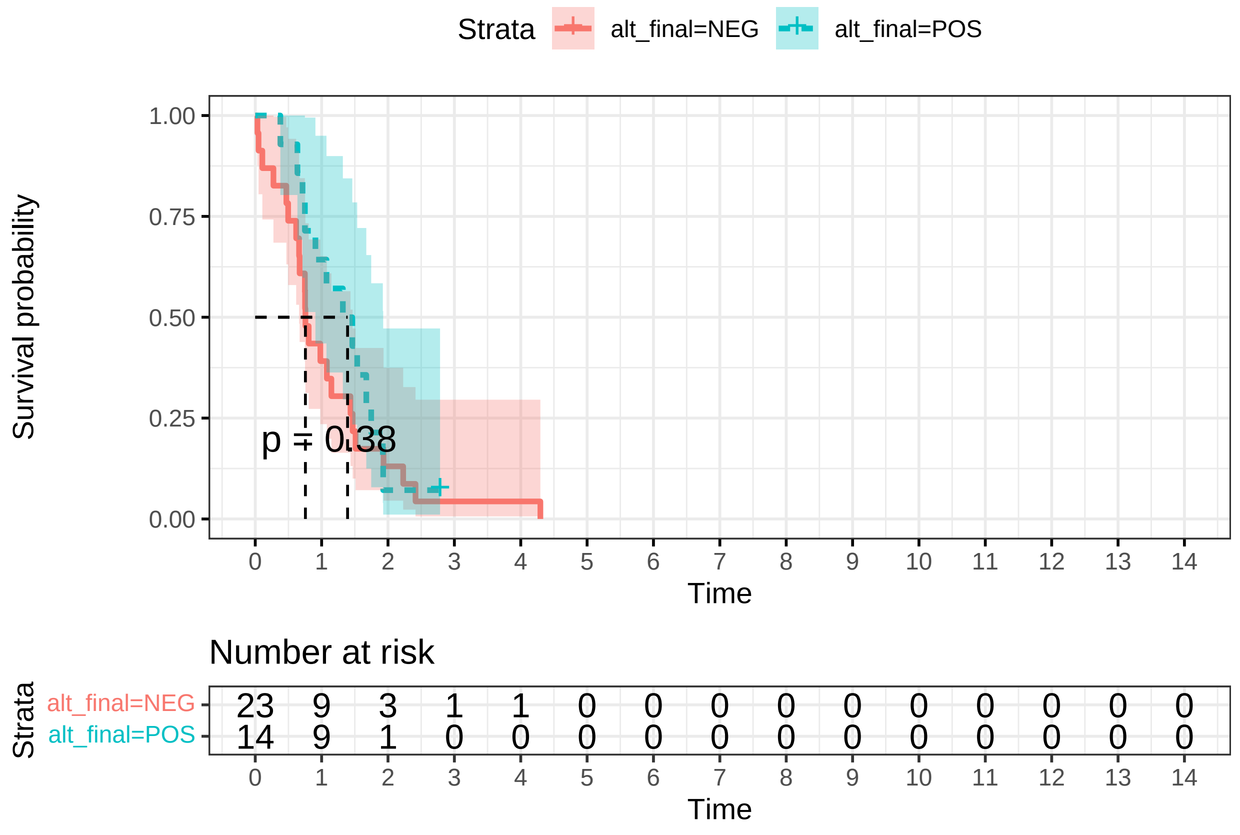
Filter to DMG only and non DMG and run KM analysis

```r
# filter samples
meta_formatted_dmg <- meta_formatted_hgat %>%
  dplyr::filter(dmg_h3k28 == "DMG")
meta_formatted_non_dmg <- meta_formatted_hgat %>%
  dplyr::filter(dmg_h3k28 == "non-DMG")
```

Run for DMG only

```r
# define model
model <- "survival::Surv(time = OS_years, event = OS_status_recoded) ~ alt_final"

######## run for DMG
# run survival analysis
fit <- survival::survdiff(formula(model),
                          data = meta_formatted_dmg)
# Obtain p value for Chi-Squared stat
fit$p.value <- pchisq(fit$chisq, df = length(fit$n) - 1, lower = FALSE)

# save the output
saveRDS(fit, file.path(output_dir, "log_rank_survival_per_alt_os_only_dmg.RDS"))
```

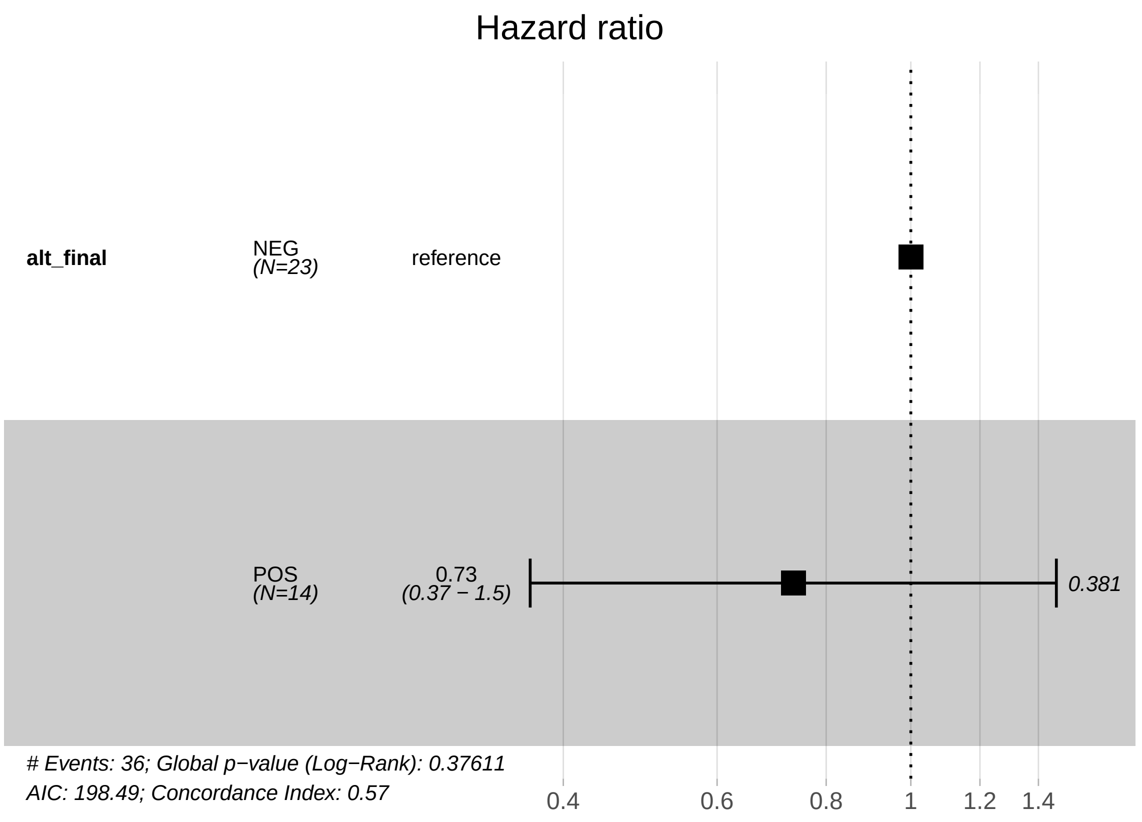
```r
# generate plots fit
fit_plot <- survfit(formula(model), data = meta_formatted_dmg)

# output the plot
plot_logrank <- survminer::ggsurvplot(fit_plot,
                                       data=meta_formatted_dmg,
                                       xlim = c(0, 14),
                                       break.time.by = 1,
                                       pval = TRUE,
                                       conf.int = TRUE,
                                       risk.table = TRUE, # Add risk table
                                       linetype = "strata", # Change line type by groups
                                       surv.median.line = "hv", # Specify median survival
                                       ggtheme = theme_bw())

print(plot_logrank)
```



```r
# Make this plot a combined plot
surv_plot_logrank <- cowplot::plot_grid(plot_logrank[[1]],
                                        plot_logrank[[2]],
                                        nrow = 2,
                                        rel_heights = c(2.5, 1))
# Save the plot
cowplot::save_plot(filename = file.path(plots_dir,
                                        "logrank_survival_by_alt_os_only_dmg.png"),
                   plot = surv_plot_logrank)

# run cox-regression
fit <- survival::coxph(
         formula(model),
         data = meta_formatted_dmg
      )
# generate output
table <- broom::tidy(fit)

# Save the table data in a TSV
readr::write_tsv(table, file.path(output_dir, "cox_reg_results_per_alt_os_only_dmg.tsv"))

print(table)
```

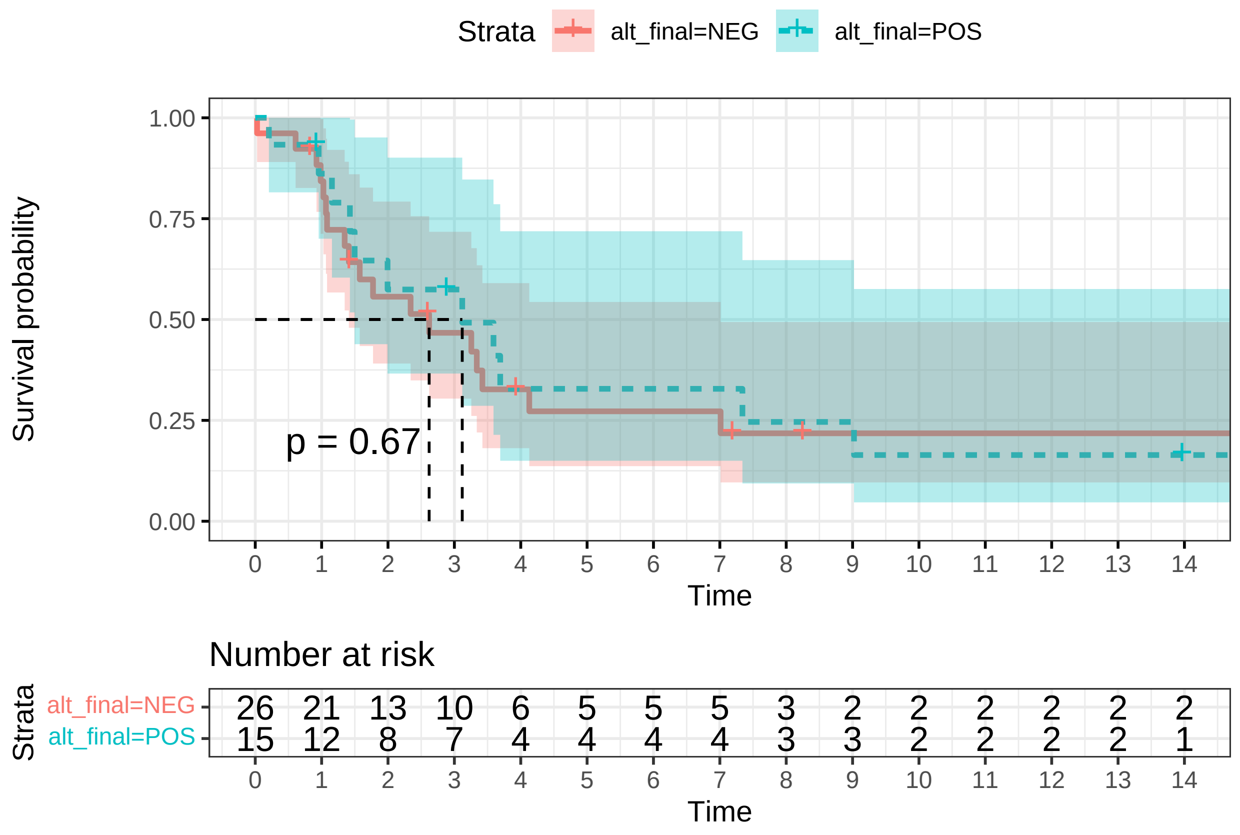
```
## # A tibble: 1 x 5
##   term         estimate std.error statistic p.value
##   <chr>           <dbl>     <dbl>     <dbl>   <dbl>
## 1 alt_finalPOS   -0.310     0.354     -0.876   0.381
```

```
# printout the plot
forest_coxph <- survminer::ggforest(fit, data = meta_formatted_dmg)
print(forest_coxph)
```



Run for non-DMG only

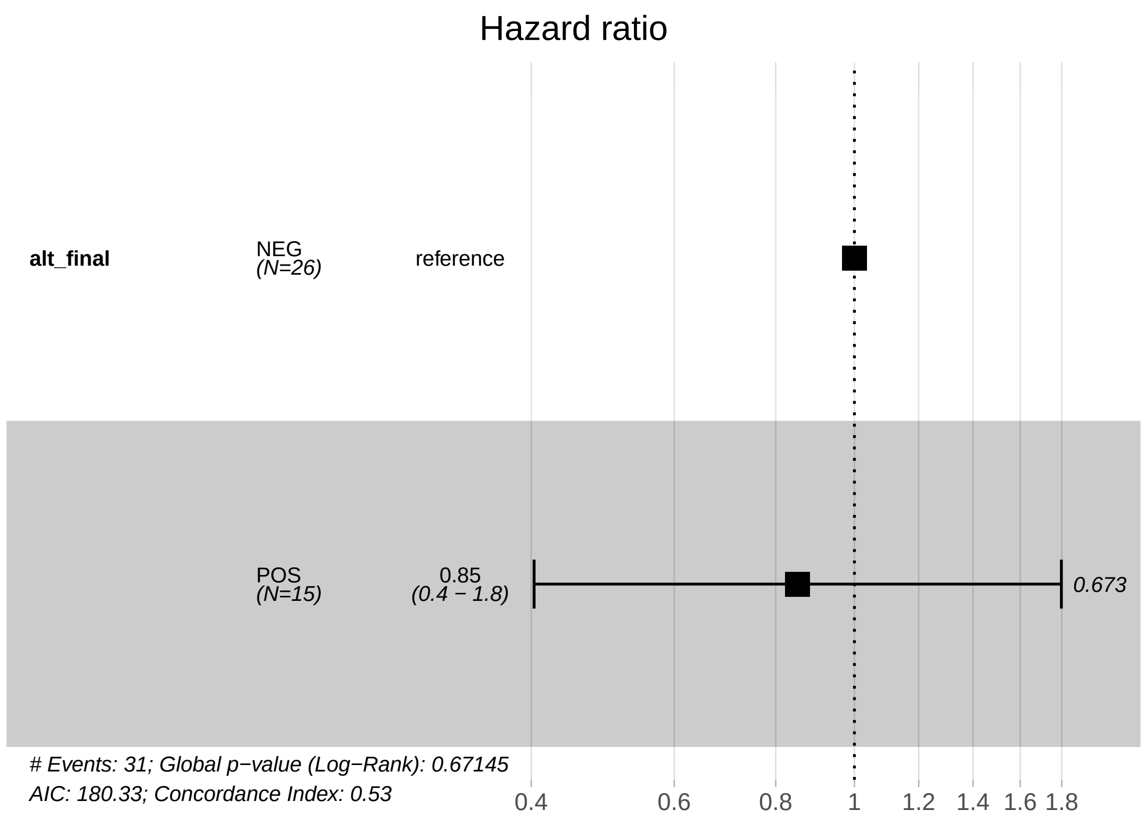
```r
######## run for non-DMG
# run survival analysis
fit <- survival::survdiff(formula(model),
                          data = meta_formatted_non_dmg)
# Obtain p value for Chi-Squared stat
fit$p.value <- pchisq(fit$chisq, df = length(fit$n) - 1, lower = FALSE)

# save the output
saveRDS(fit, file.path(output_dir, "log_rank_survival_per_alt_os_non_dmg.RDS"))

# generate plots fit
fit_plot <- survfit(formula(model), data = meta_formatted_non_dmg)

# output the plot
plot_logrank <- survminer::ggsurvplot(fit_plot,
                                      data=meta_formatted_non_dmg,
                                      xlim = c(0, 14),
                                      break.time.by = 1,
                                      pval = TRUE,
                                      conf.int = TRUE,
                                      risk.table = TRUE, # Add risk table
                                      linetype = "strata", # Change line type by groups
                                      surv.median.line = "hv", # Specify median survival
                                      ggtheme = theme_bw())

print(plot_logrank)
```



```
# Make this plot a combined plot
surv_plot_logrank <- cowplot::plot_grid(plot_logrank[[1]],
                                        plot_logrank[[2]],
                                        nrow = 2,
                                        rel_heights = c(2.5, 1))
# Save the plot
cowplot::save_plot(filename = file.path(plots_dir,
                                        "logrank_survival_by_alt_os_only_non_dmg.png"),
                   plot = surv_plot_logrank)

# run cox-regression
fit <- survival::coxph(
        formula(model),
        data = meta_formatted_non_dmg
      )
# generate output
table <- broom::tidy(fit)

# Save the table data in a TSV
readr::write_tsv(table, file.path(output_dir, "cox_reg_results_per_alt_os_only_non_dmg.tsv"))

print(table)
```

```
## # A tibble: 1 x 5
##   term          estimate std.error statistic p.value
##   <chr>            <dbl>     <dbl>     <dbl>   <dbl>
## 1 alt_finalPOS   -0.161     0.381    -0.421   0.673
```

```
# printout the plot
forest_coxph <- survminer::ggforest(fit, data = meta_formatted_non_dmg)
print(forest_coxph)
```


Hazard ratio
alt_final
NEG (N=26)
reference
POS (N=15)
0.85 (0.4 − 1.8)
# Events: 31; Global p−value (Log−Rank): 0.67145
AIC: 180.33; Concordance Index: 0.53