

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 + ALT vs. non-ALT + sex + ATRX (mut N/Y)
- 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)

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

```

root_dir <- rprojroot::find_root(rprojroot::has_dir(".git"))
analysis_dir <- file.path(root_dir, "analyses", "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

```

# get the meta information
meta <- readr::read_tsv(file.path(root_dir,
                                "analyses/add-histologies/output/ALT PBTA oct 2021 (including all plates)-up
                                # remove existing ones to get newer data
                                dplyr::select(-c("OS_days", "OS_status"))

```

```
## Rows: 900 Columns: 115
```

```

## -- Column specification -----
## Delimiter: "\t"
## chr (56): 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 (7): ATRX Stata, cell_line_composition, DAXX_fusion, ...34, NF...39, CC...

##
## 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.

```

```

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

```

```
## Rows: 2840 Columns: 38
```

```

## -- Column specification -----
## Delimiter: "\t"
## chr (33): Kids_First_Biospecimen_ID, sample_id, aliquot_id, Kids_First_Part...
## 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

```
# 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"
```

```
meta$PFS_days <- as.numeric(meta$PFS_days)
```

```
## Warning: NAs introduced by coercion
```

```
# 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)) %>%
  # calculate the years
  dplyr::mutate(OS_years = OS_days / 365.25) %>%
  dplyr::mutate(PFS_years = PFS_days / 365.25) %>%
  dplyr::mutate(PFS_status = if_else(PFS_days < OS_days, 1, 0)) %>%
  # 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(dmng_h3k28 = case_when(
    grepl("DMG, H3 K28", molecular_subtype) ~ "DMG",
    TRUE ~ "non-DMG"
  )) %>%
  # categorize ATRX
  dplyr::mutate(atr_x_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`)

# define as factor
meta_formatted$dmng_h3k28 <- factor(meta_formatted$dmng_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$phenotype <- factor(meta_formatted$phenotype, levels = c("non-ALT", "ALT"))
```

```

meta_formatted$atrx_mut <- factor(meta_formatted$atrx_mut, levels = c("non_ATRX_mut", "ATRX_mut"))

# define as numeric
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", "phenotype")){
  # 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

  # run survival analysis
  fit <- survival::survdif(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, ".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,

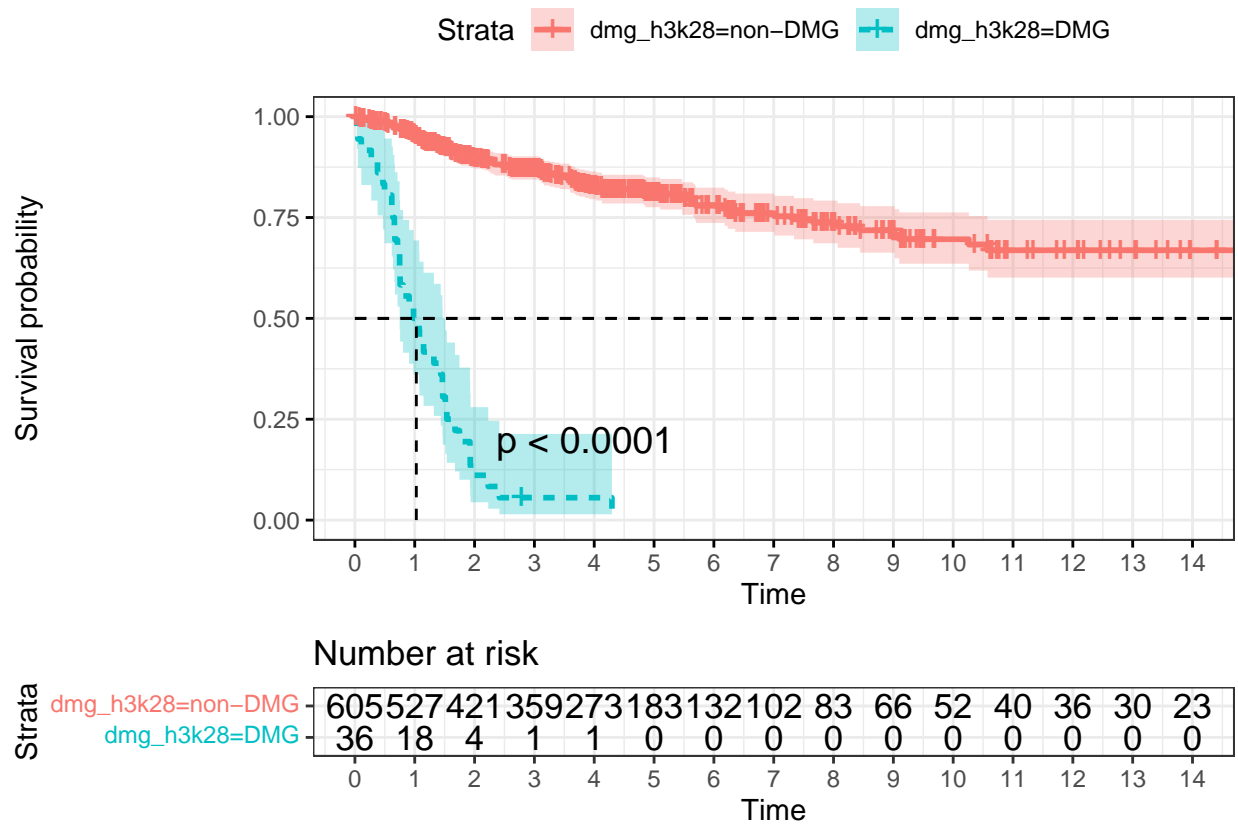
```

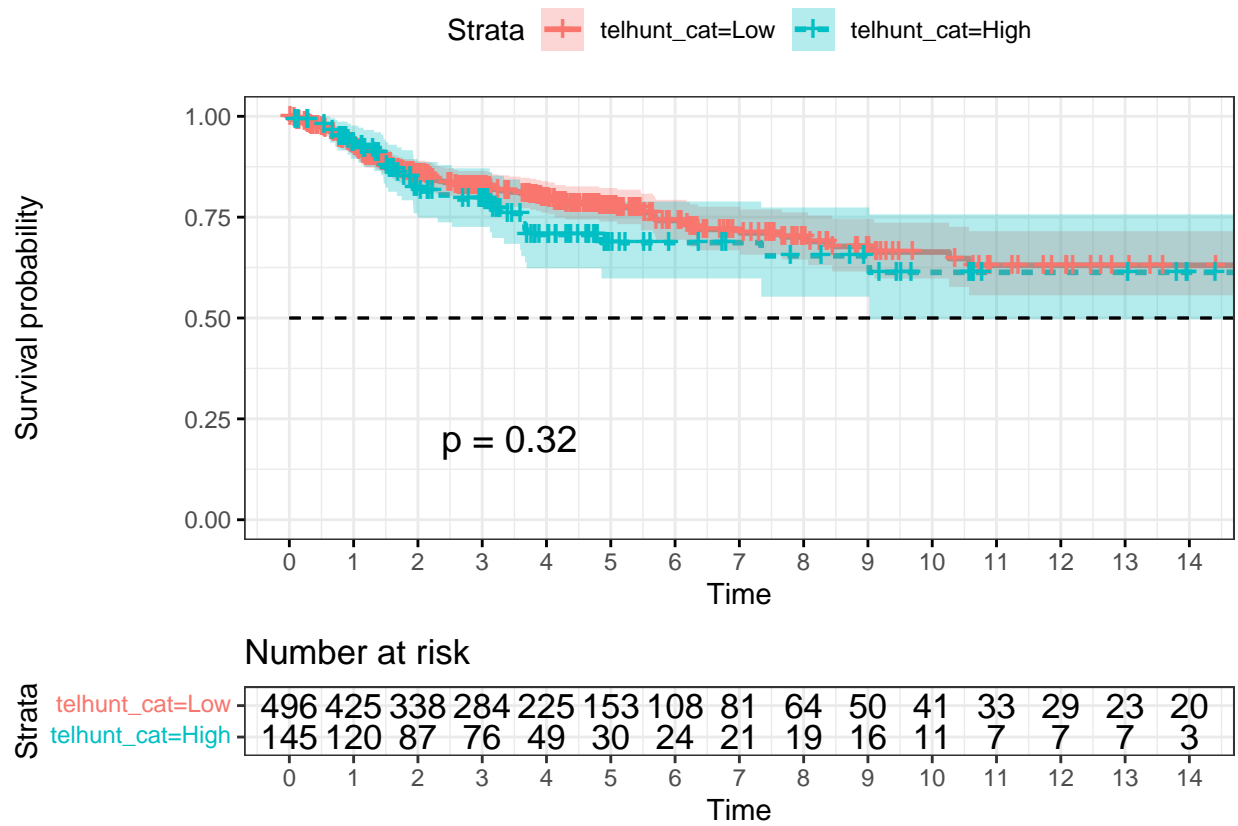
```

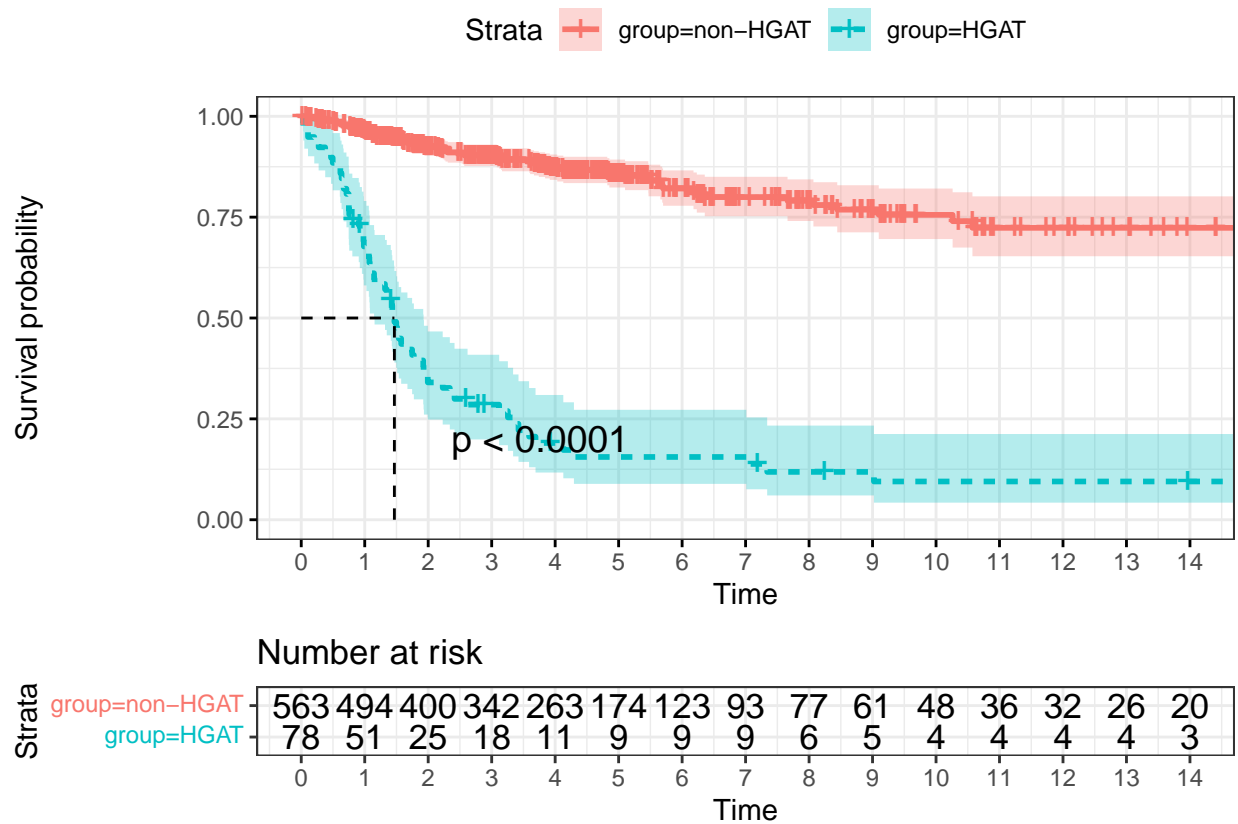
rel_heights = c(2.5, 1))

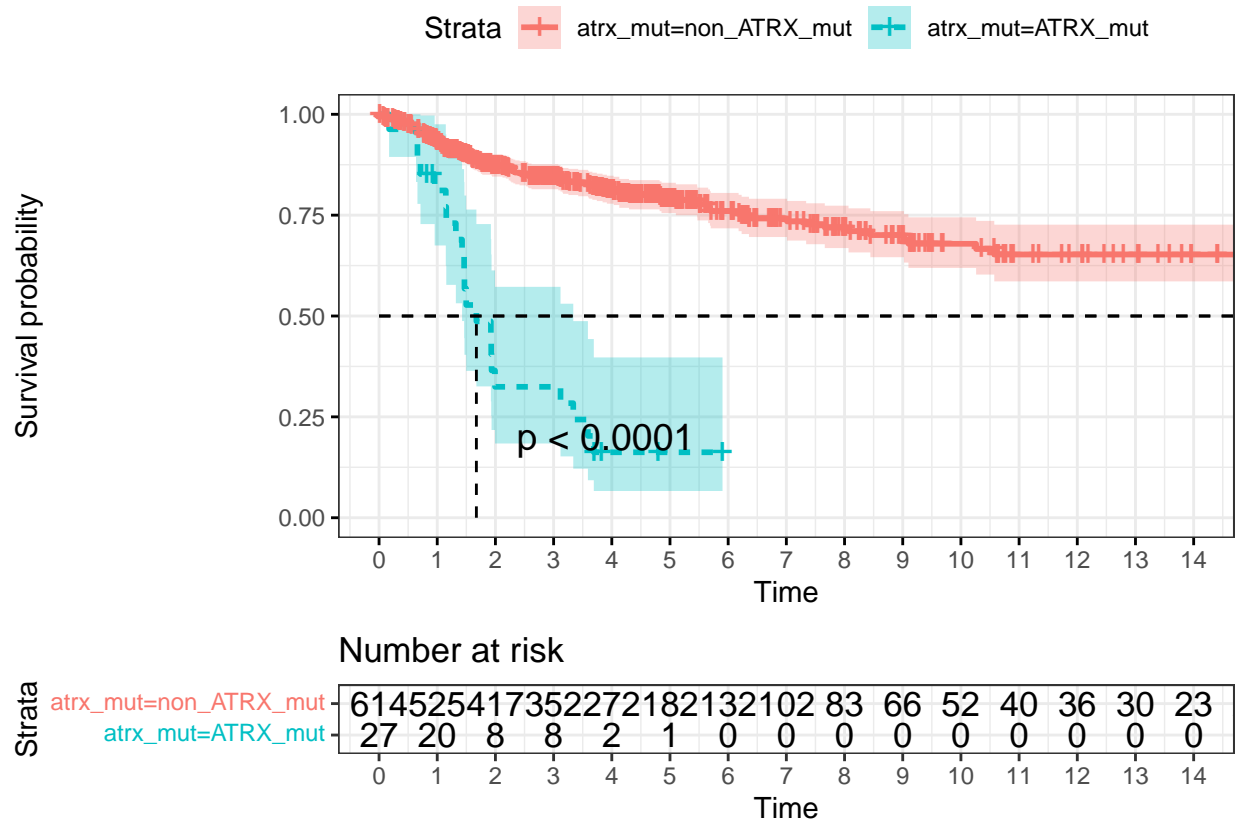
# Save the plot
cowplot::save_plot(filename = file.path(plots_dir,
                                         paste0("logrank_survival_by_", ind_var, ".png")),
                    plot = surv_plot_logrank)
}

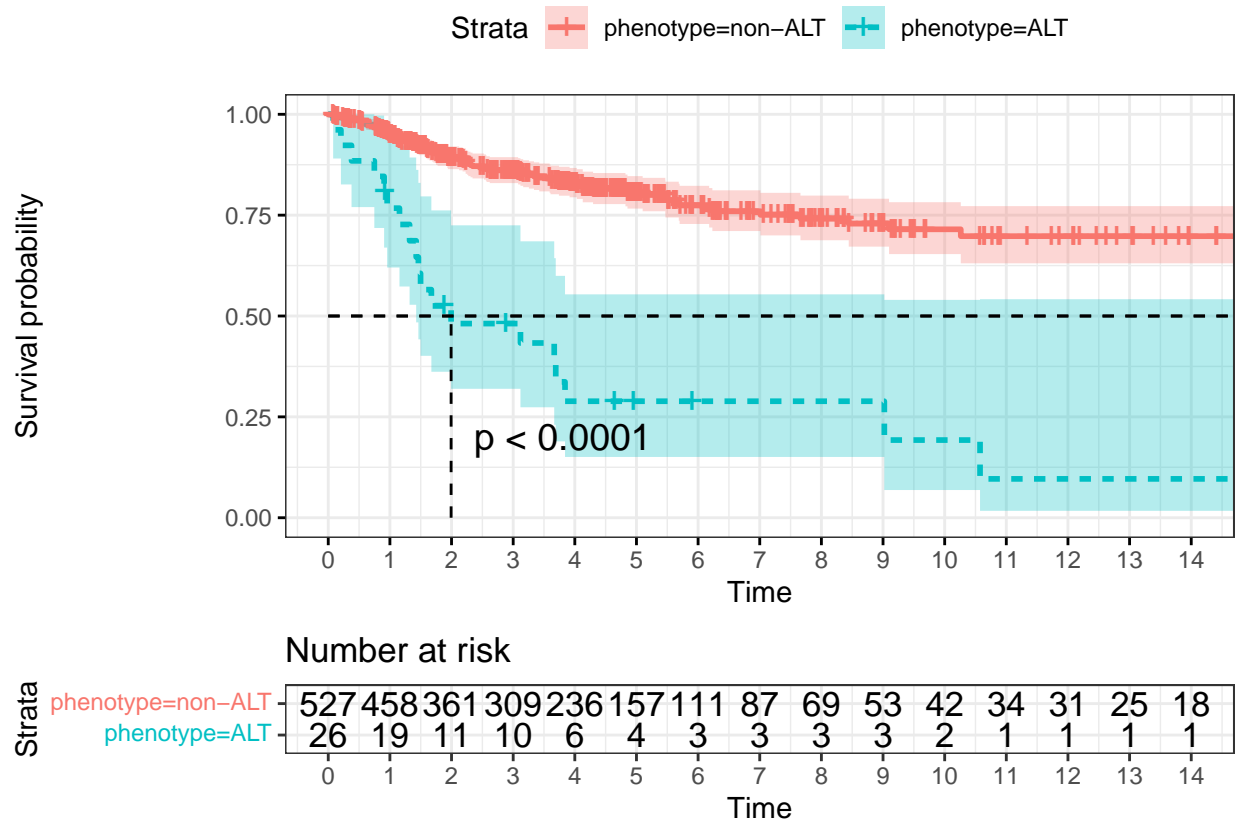
```











Multivariate analysis

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

```
# define multi-variables that we are using for analyzing survival
list_of_variates <- c("dmg_h3k28+group+telhunt_cat+reported_gender+atr_x_mut",
                      "dmg_h3k28+group+telhunt_score+reported_gender+atr_x_mut",
                      "dmg_h3k28+group+phenotype+reported_gender+atr_x_mut"
)

# 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)

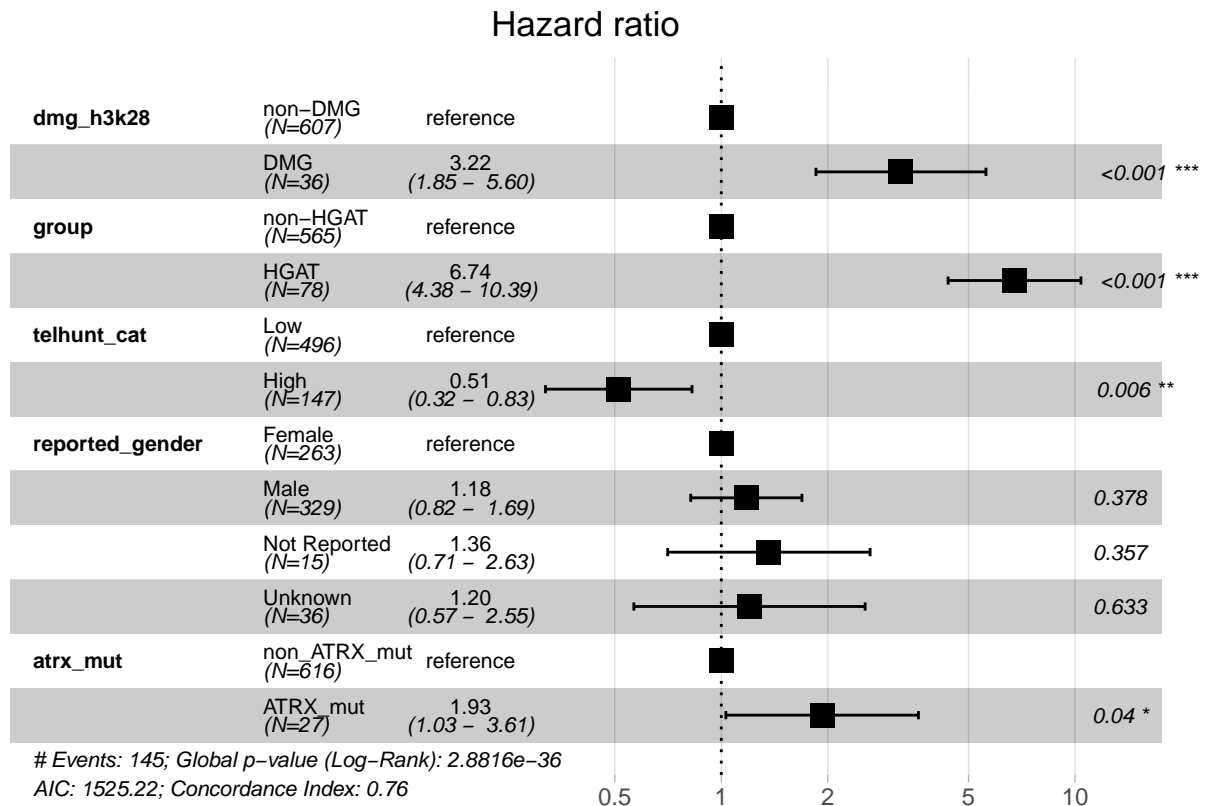
# printout the plot
forest_coxph <- survminer::ggforest(fit, data = data_used)
print(forest_coxph)
}

```

```

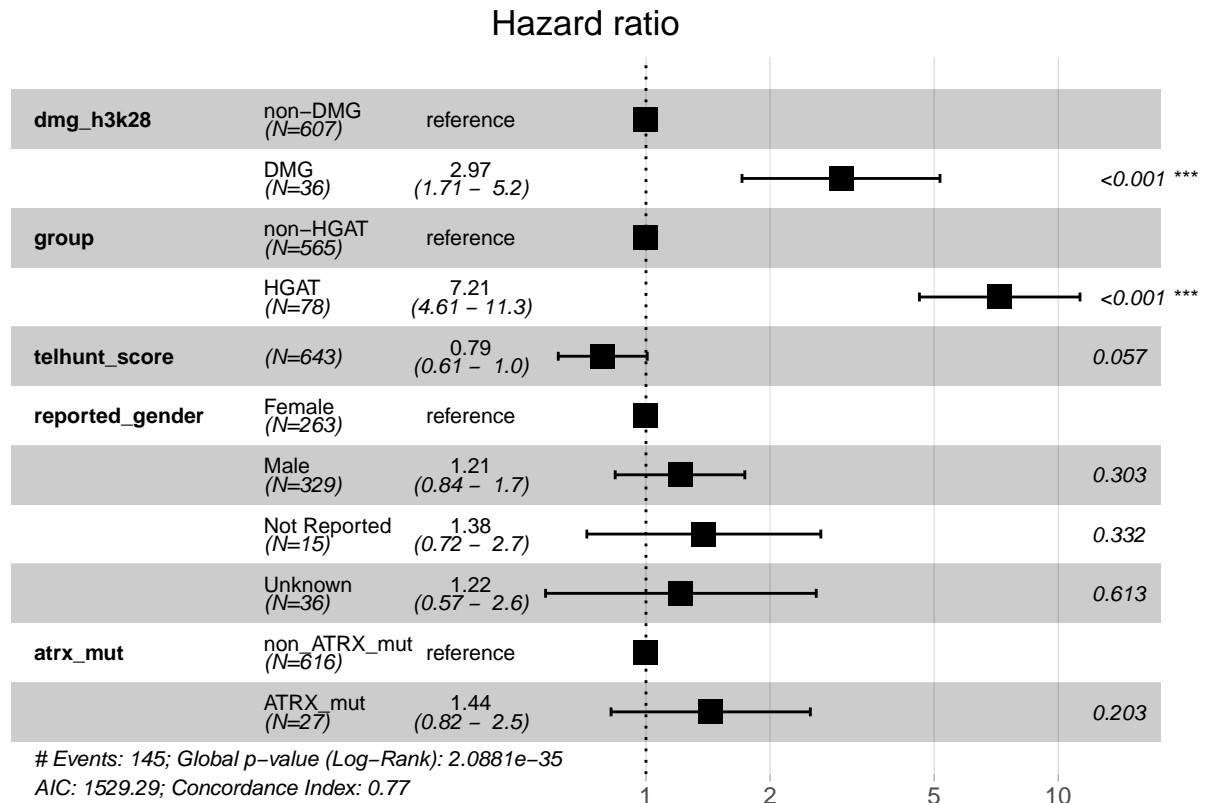
## # A tibble: 7 x 5
##   term                                estimate std.error statistic  p.value
##   <chr>                                <dbl>    <dbl>    <dbl>    <dbl>
## 1 dmg_h3k28DMG                        1.17      0.282     4.14 3.43e- 5
## 2 groupHGAT                          1.91      0.221     8.65 5.08e-18
## 3 telhunt_catHigh                    -0.668     0.243    -2.74 6.06e- 3
## 4 reported_genderMale                 0.163     0.185     0.882 3.78e- 1
## 5 reported_genderNot Reported         0.310     0.336     0.921 3.57e- 1
## 6 reported_genderUnknown              0.184     0.384     0.478 6.33e- 1
## 7 atrx_mutATRX_mut                   0.656     0.320     2.05 4.01e- 2

```

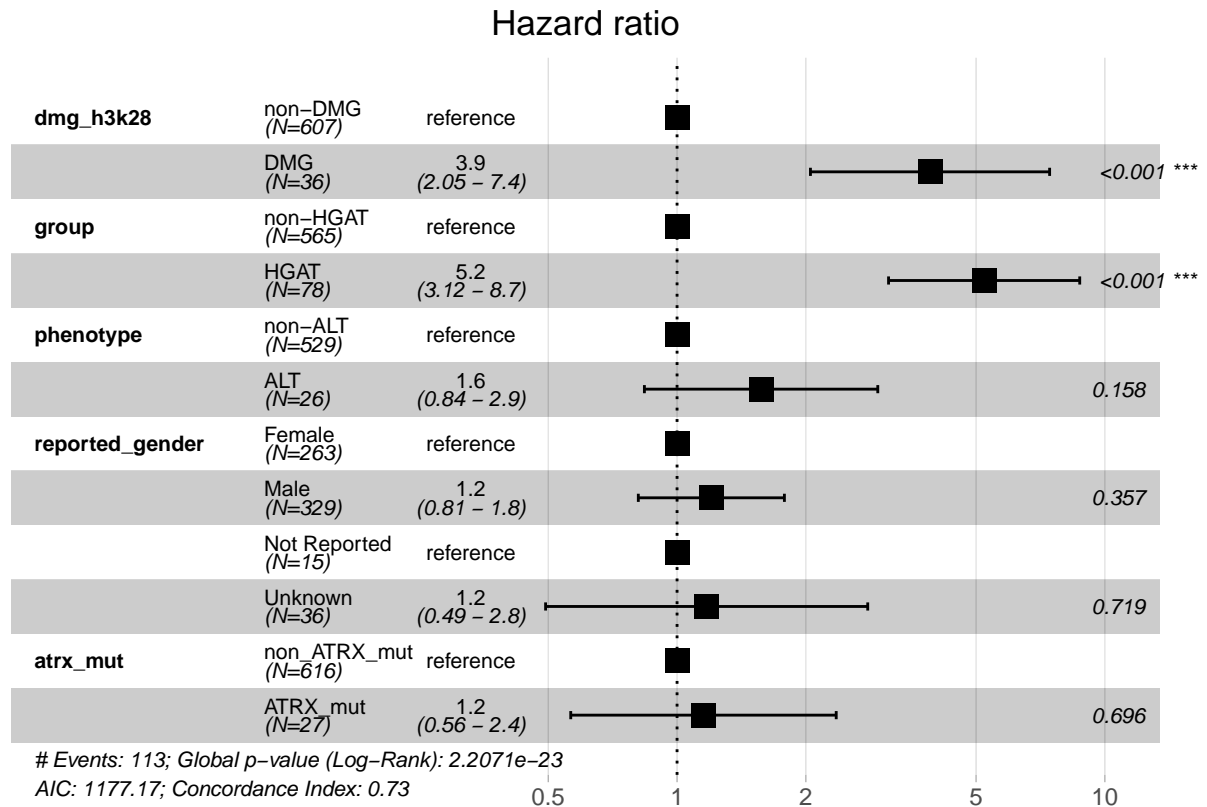


```
## # A tibble: 7 x 5
```

##	term	estimate	std.error	statistic	p.value
##	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
## 1	dmg_h3k28DMG	1.09	0.282	3.86	1.13e- 4
## 2	groupHGAT	1.98	0.229	8.64	5.86e-18
## 3	telhunt_score	-0.241	0.127	-1.90	5.68e- 2
## 4	reported_genderMale	0.190	0.185	1.03	3.03e- 1
## 5	reported_genderNot Reported	0.323	0.333	0.969	3.32e- 1
## 6	reported_genderUnknown	0.195	0.386	0.506	6.13e- 1
## 7	atrx_mutATRX_mut	0.362	0.284	1.27	2.03e- 1



```
## # A tibble: 6 x 5
##   term                estimate std.error statistic  p.value
##   <chr>              <dbl>    <dbl>    <dbl>    <dbl>
## 1 dmg_h3k28DMG       1.36      0.328     4.15 3.38e- 5
## 2 groupHGAT          1.65      0.262     6.30 3.04e-10
## 3 phenotypeALT       0.452     0.320     1.41 1.58e- 1
## 4 reported_genderMale 0.185     0.201     0.920 3.57e- 1
## 5 reported_genderUnknown 0.159     0.442     0.360 7.19e- 1
## 6 atrx_mutATRX_mut   0.142     0.364     0.390 6.96e- 1
```



B run for HGAT only

Univariate analysis

```
for(ind_var in c("dmg_h3k28", "telhunt_cat", "atr_x_mut", "phenotype")){
  # 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)
```

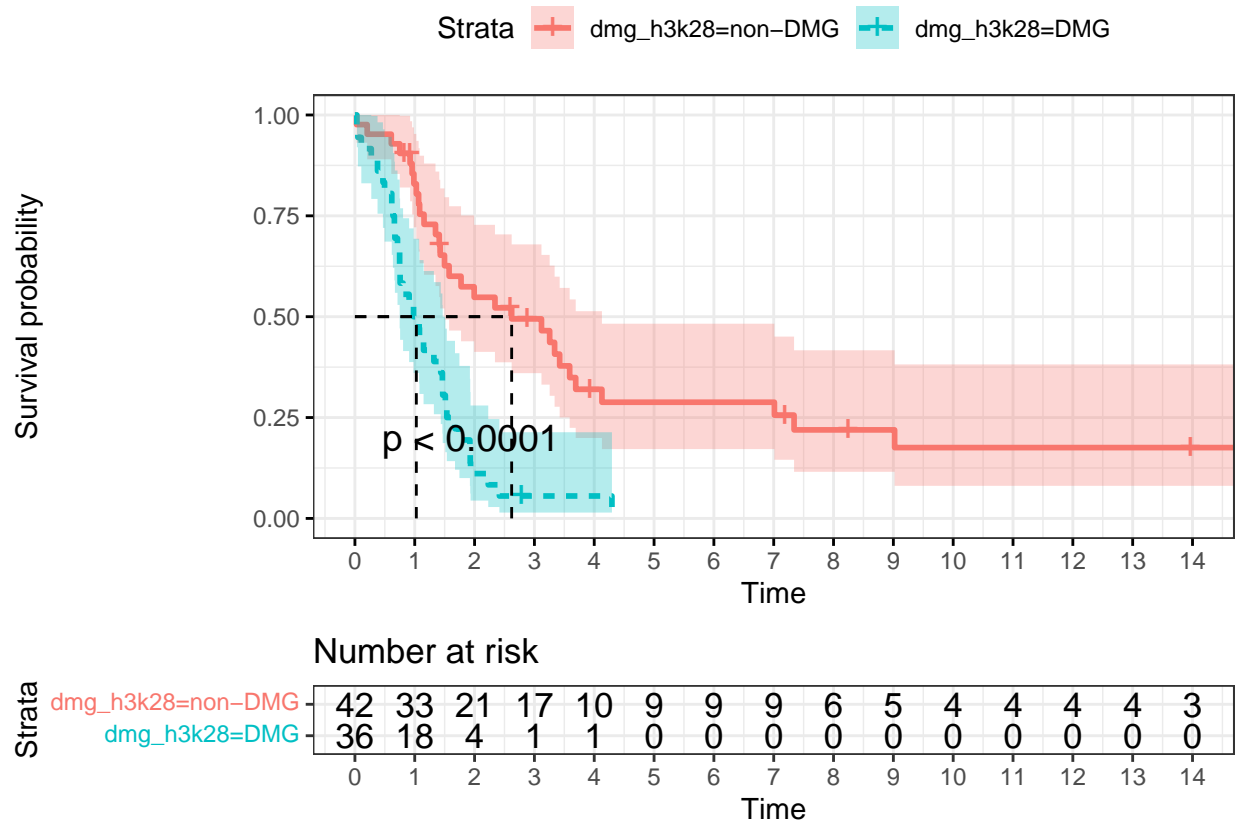
```

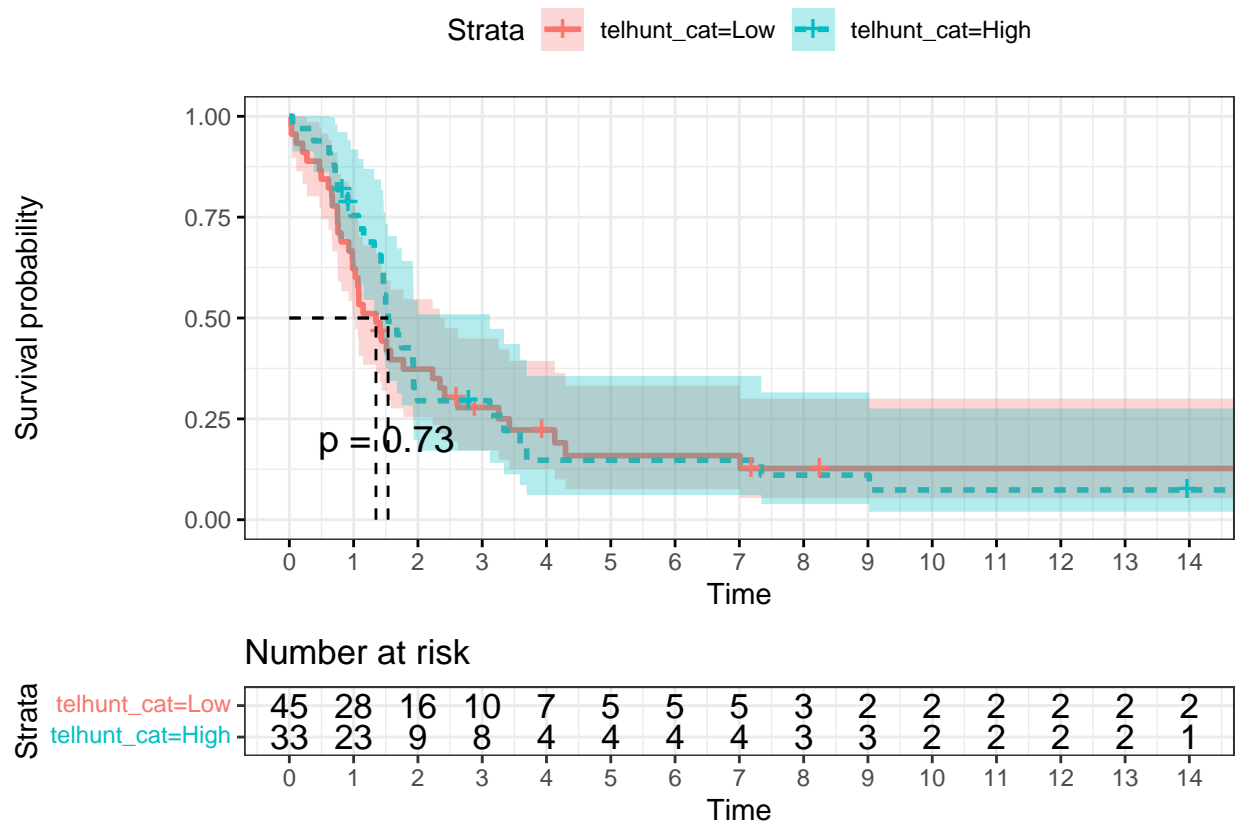
# output the plot
plot_logrank <- survminer::ggsurvplot(fit_plot,
                                     data=data_used,
                                     xlim = c(0, 14),
                                     break.time.by = 1,
                                     pval = TRUE,
                                     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())

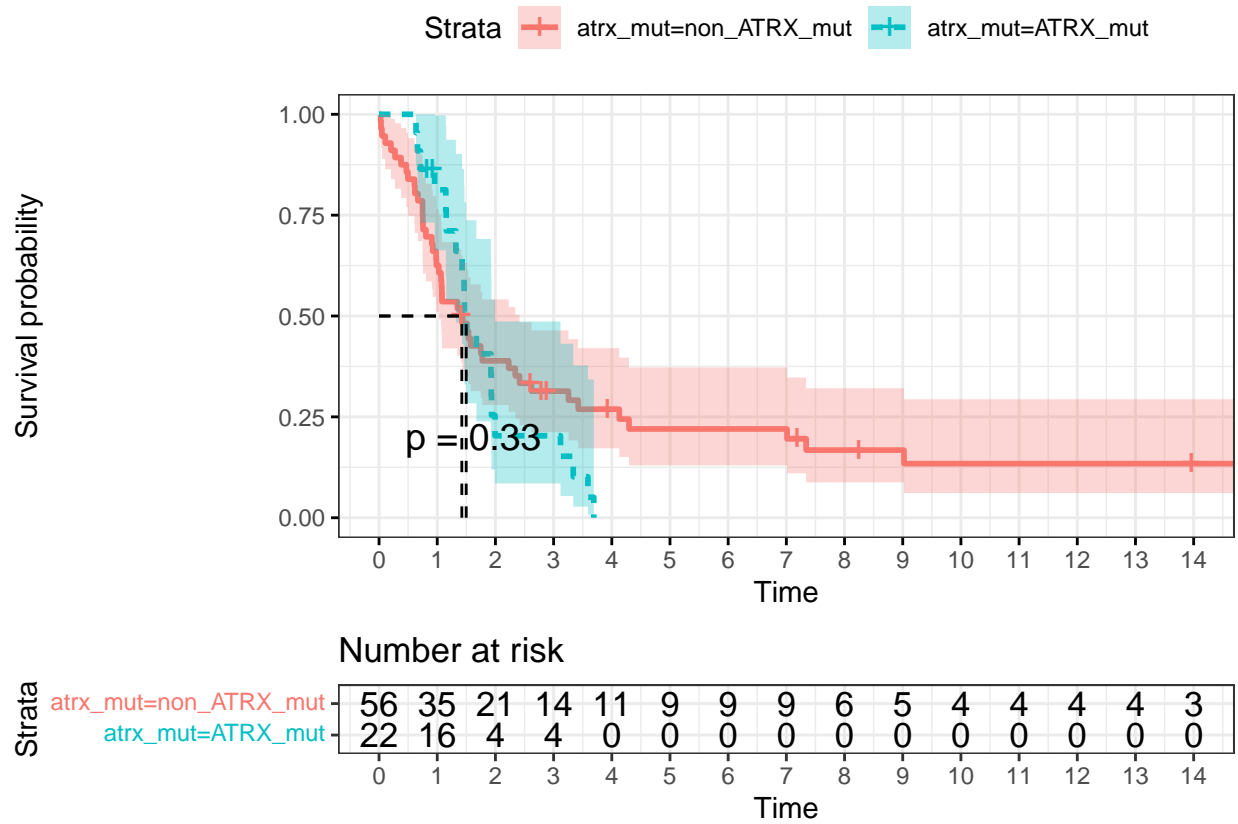
print(plot_logrank)
# Make this plot a combined plot
surv_plot_logrank <- cowplot::plot_grid(plot_logrank[[1]],
                                       plot_logrank[[2]],
                                       nrow = 2,
                                       rel_heights = c(2.5, 1))

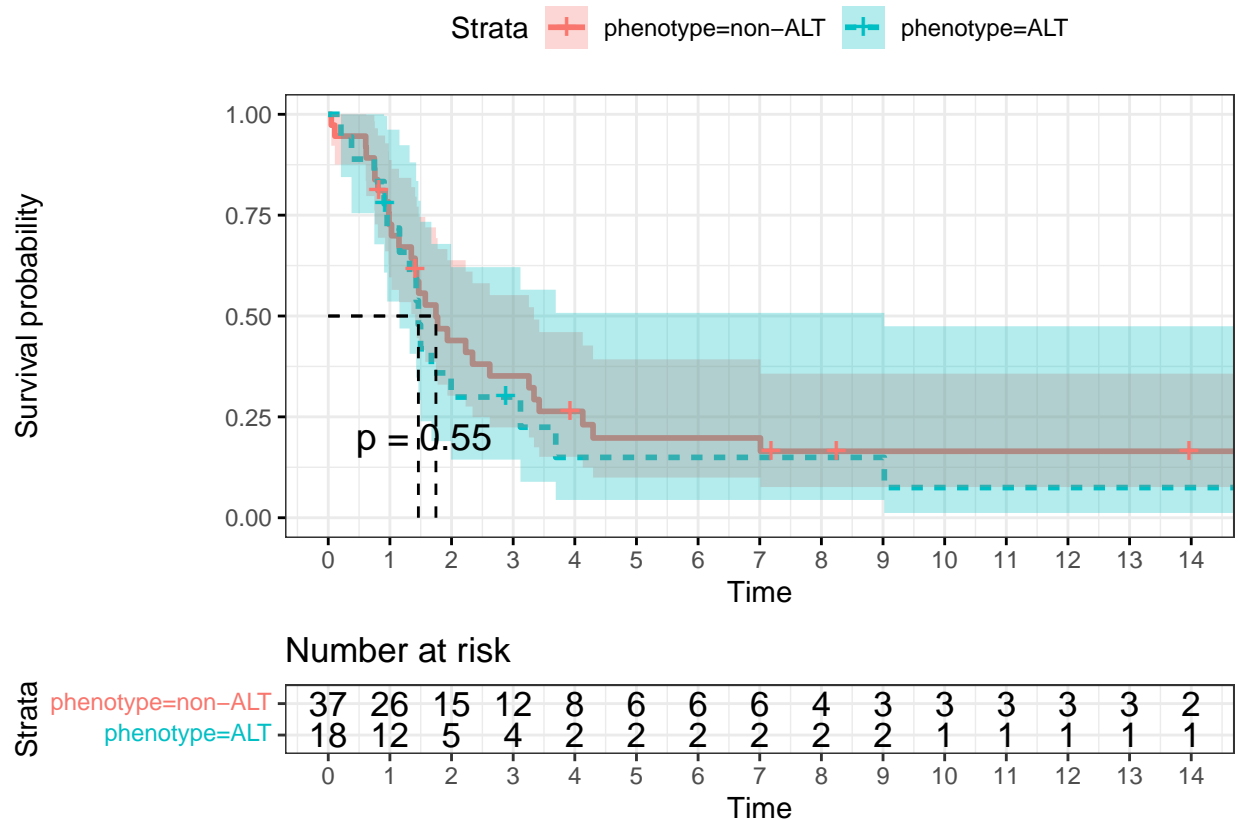
# Save the plot
cowplot::save_plot(filename = file.path(plots_dir,
                                       paste0("logrank_survival_by_", ind_var, "_os_only_hgat.png")),
                  plot = surv_plot_logrank)
}

```









Multivariate analysis

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

```
# define multi-variables that we are using for analyzing survival
list_of_variates <- c("dmg_h3k28+telhunt_cat+reported_gender+atrx_mut",
                      "dmg_h3k28+telhunt_score+reported_gender+atrx_mut",
                      "dmg_h3k28+phenotype+reported_gender+atrx_mut"
)

# 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)

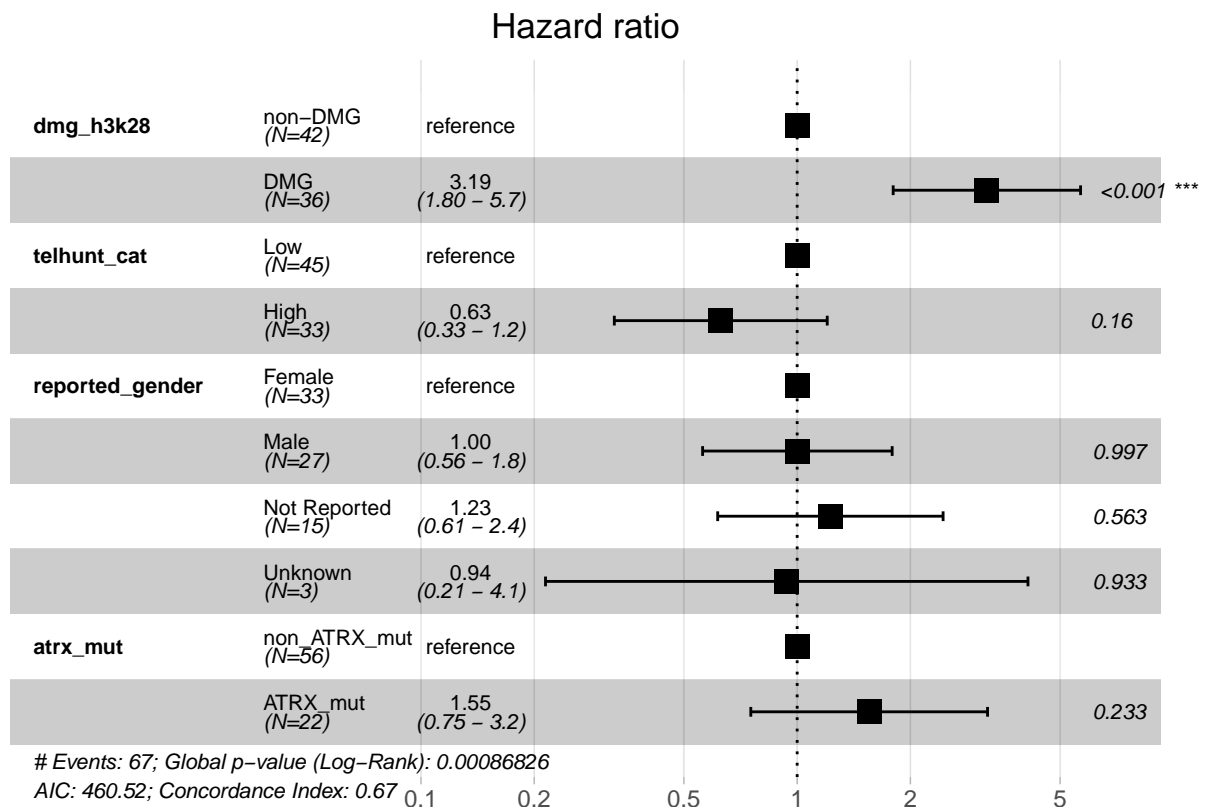
# printout the plot
forest_coxph <- survminer::ggforest(fit, data = data_used)
print(forest_coxph)
}

```

```

## # A tibble: 6 x 5
##   term                                estimate std.error statistic    p.value
##   <chr>                                <dbl>     <dbl>     <dbl>    <dbl>
## 1 dmg_h3k28DMG                        1.16       0.293     3.97  0.0000724
## 2 telhunt_catHigh                     -0.468     0.332    -1.41   0.160
## 3 reported_genderMale                 0.00129    0.296    0.00437 0.997
## 4 reported_genderNot Reported         0.203      0.352    0.578   0.563
## 5 reported_genderUnknown              -0.0638    0.753   -0.0846 0.933
## 6 atrx_mutATRX_mut                   0.441      0.370    1.19    0.233

```

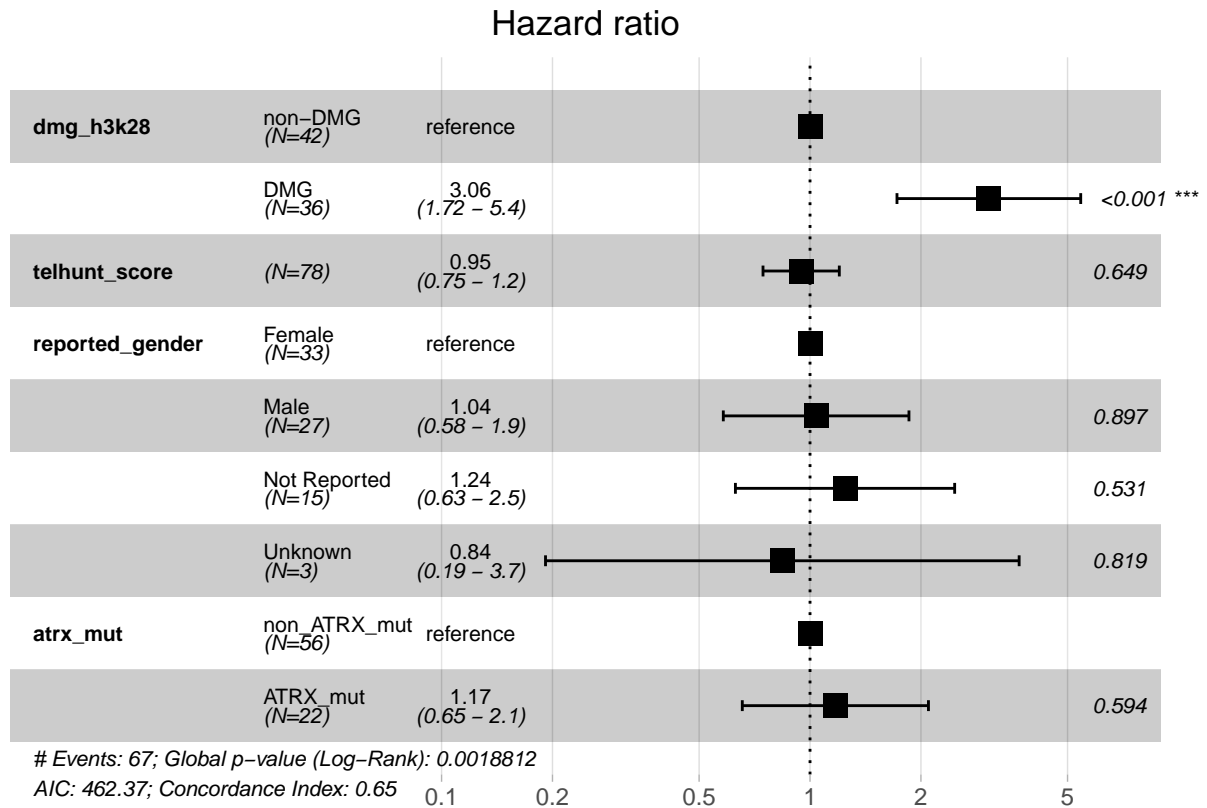


```

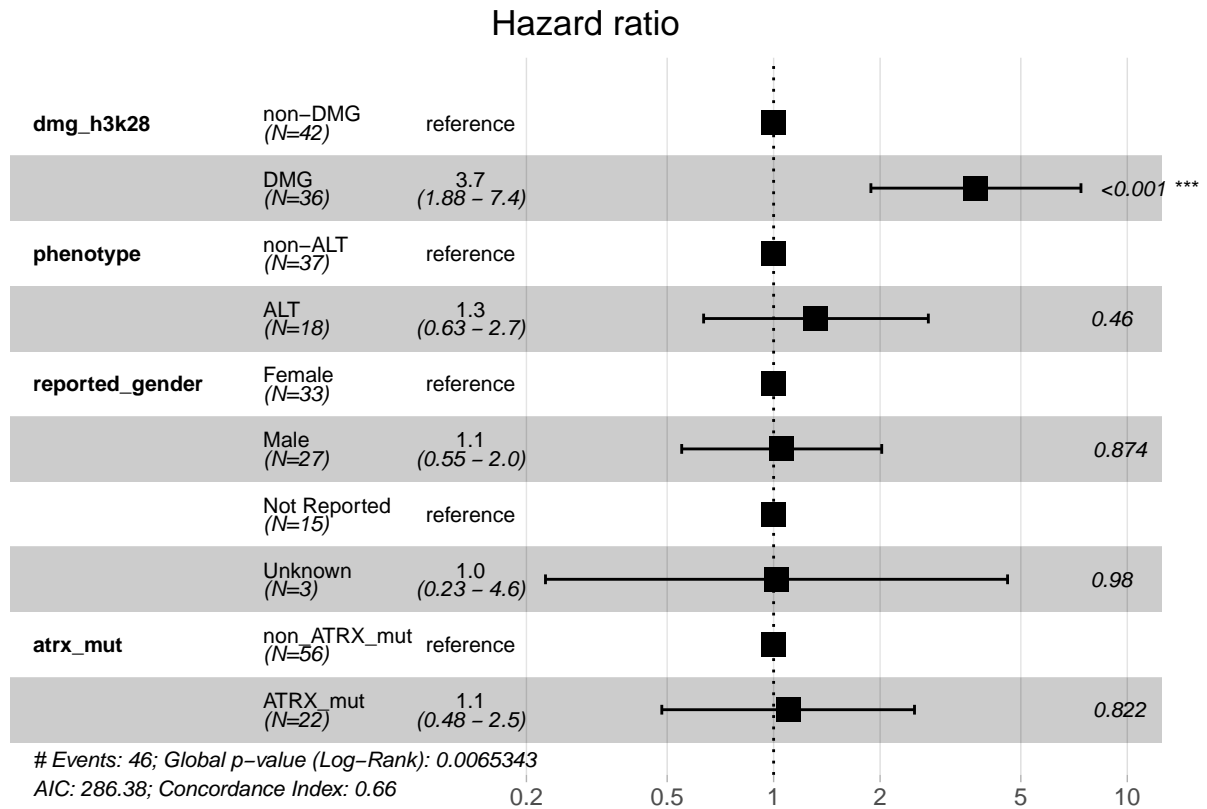
## # A tibble: 6 x 5
##   term                                estimate std.error statistic    p.value
##   <chr>                                <dbl>     <dbl>     <dbl>    <dbl>

```

```
## 1 dmg_h3k28DMG          1.12      0.293      3.82  0.000136
## 2 telhunt_score        -0.0554     0.122     -0.455 0.649
## 3 reported_genderMale    0.0382     0.296      0.129 0.897
## 4 reported_genderNot Reported  0.219     0.350      0.626 0.531
## 5 reported_genderUnknown -0.173     0.755     -0.229 0.819
## 6 atrx_mutATRX_mut      0.158     0.297      0.534 0.594
```



```
## # A tibble: 5 x 5
##   term          estimate std.error statistic  p.value
##   <chr>          <dbl>    <dbl>    <dbl>    <dbl>
## 1 dmg_h3k28DMG      1.32      0.349      3.78  0.000160
## 2 phenotypeALT      0.276     0.373      0.739  0.460
## 3 reported_genderMale 0.0527    0.332      0.159  0.874
## 4 reported_genderUnknown 0.0189    0.767      0.0246 0.980
## 5 atr_x_mutATRX_mut  0.0946    0.420      0.225  0.822
```



C run for HGAT only but use PFS

```
for(ind_var in c("dmg_h3k28", "telhunt_cat", "atrx_mut", "phenotype")){
  # define model
  model <- paste0("survival::Surv(time = PFS_years, event = PFS_status) ~ ", 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, "_pfs_only_hgat.RDS")))

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

  # output the plot
  plot_logrank <- survminer::ggsurvplot(fit_plot,
```

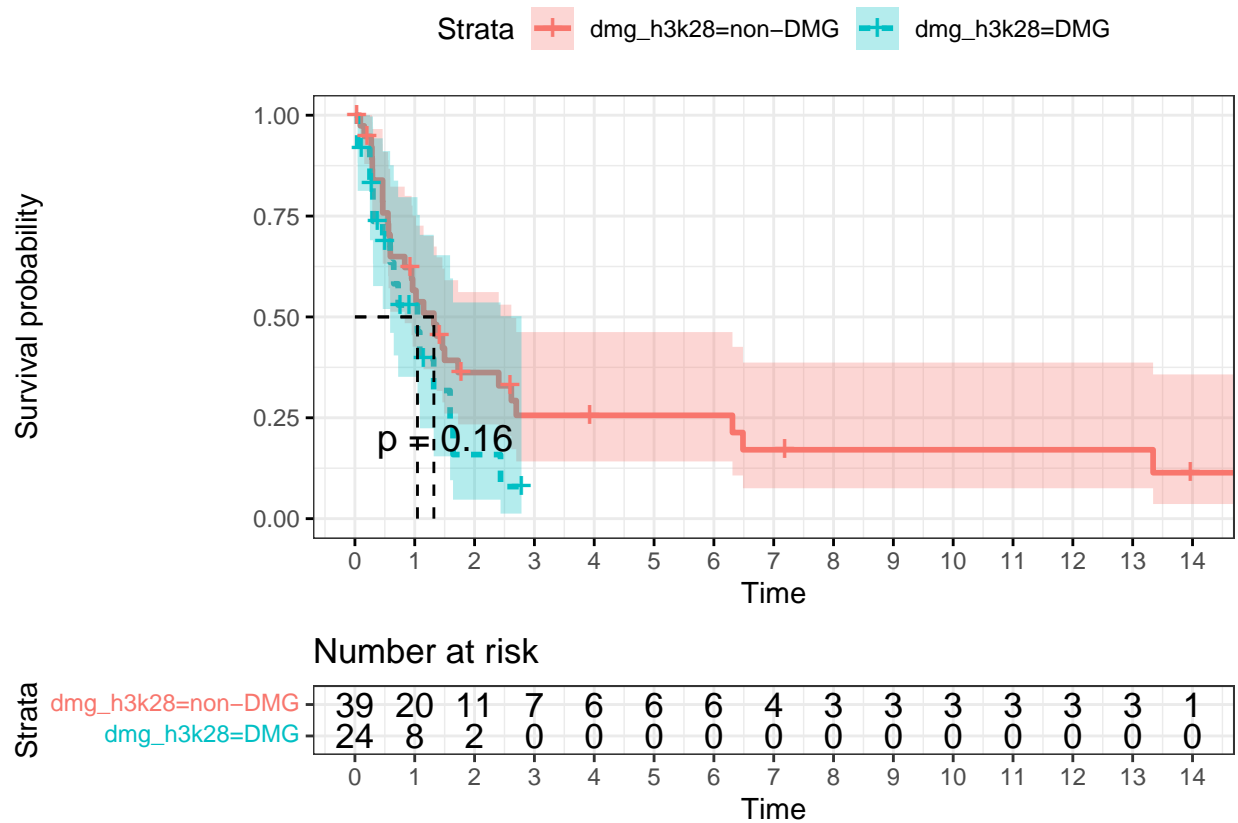
```

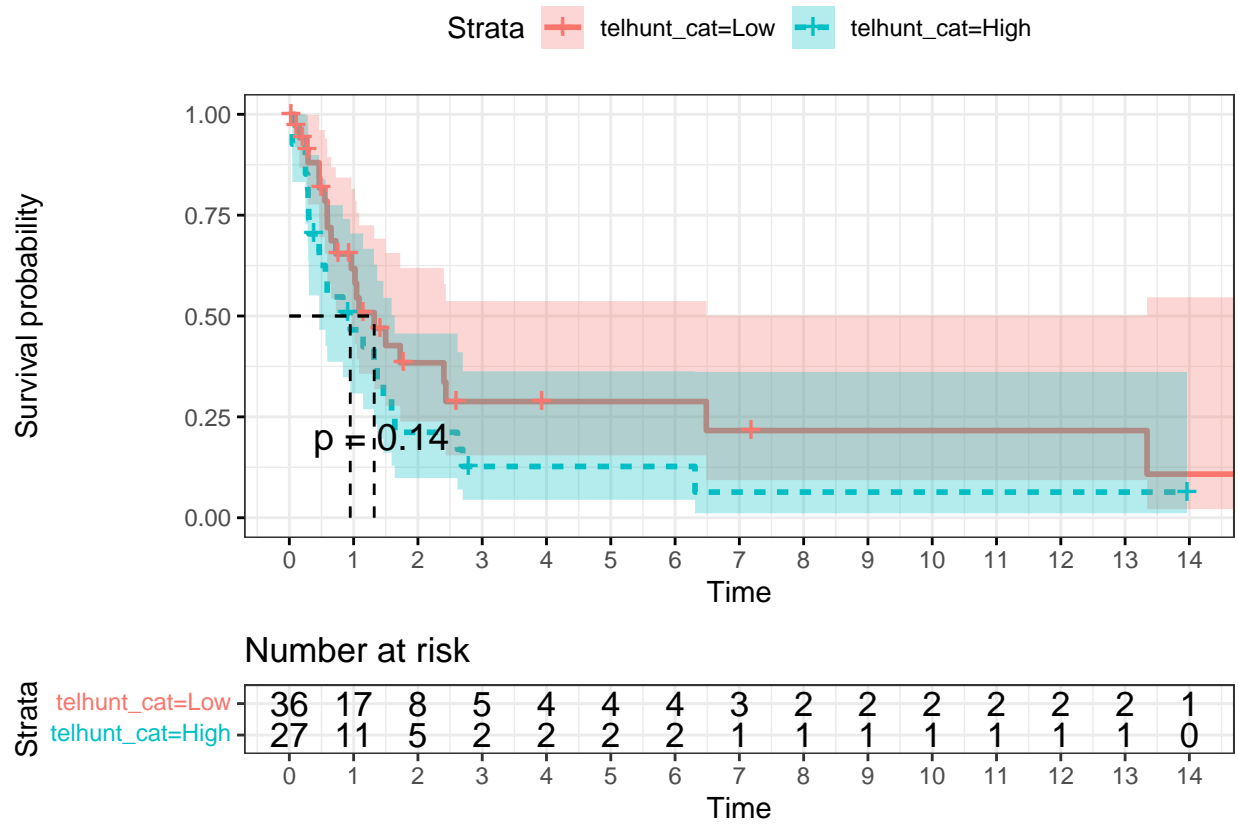
data=data_used,
xlim = c(0, 14),
break.time.by = 1,
pval = TRUE,
conf.int = TRUE,
risk.table = TRUE, # Add risk table
linetype = "strata", # Change line type by groups
surv.median.line = "hv", # Specify median survival
ggtheme = theme_bw())

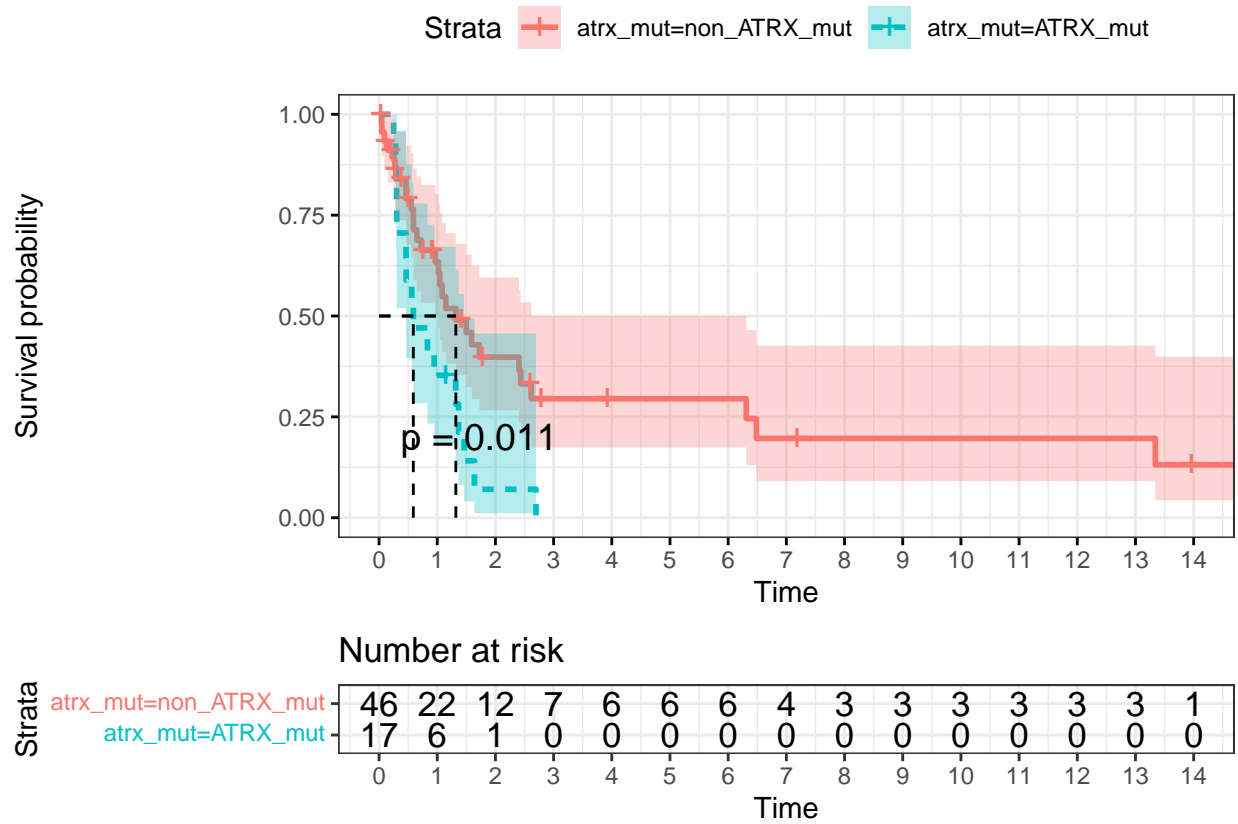
print(plot_logrank)
# Make this plot a combined plot
surv_plot_logrank <- cowplot::plot_grid(plot_logrank[[1]],
                                       plot_logrank[[2]],
                                       nrow = 2,
                                       rel_heights = c(2.5, 1))

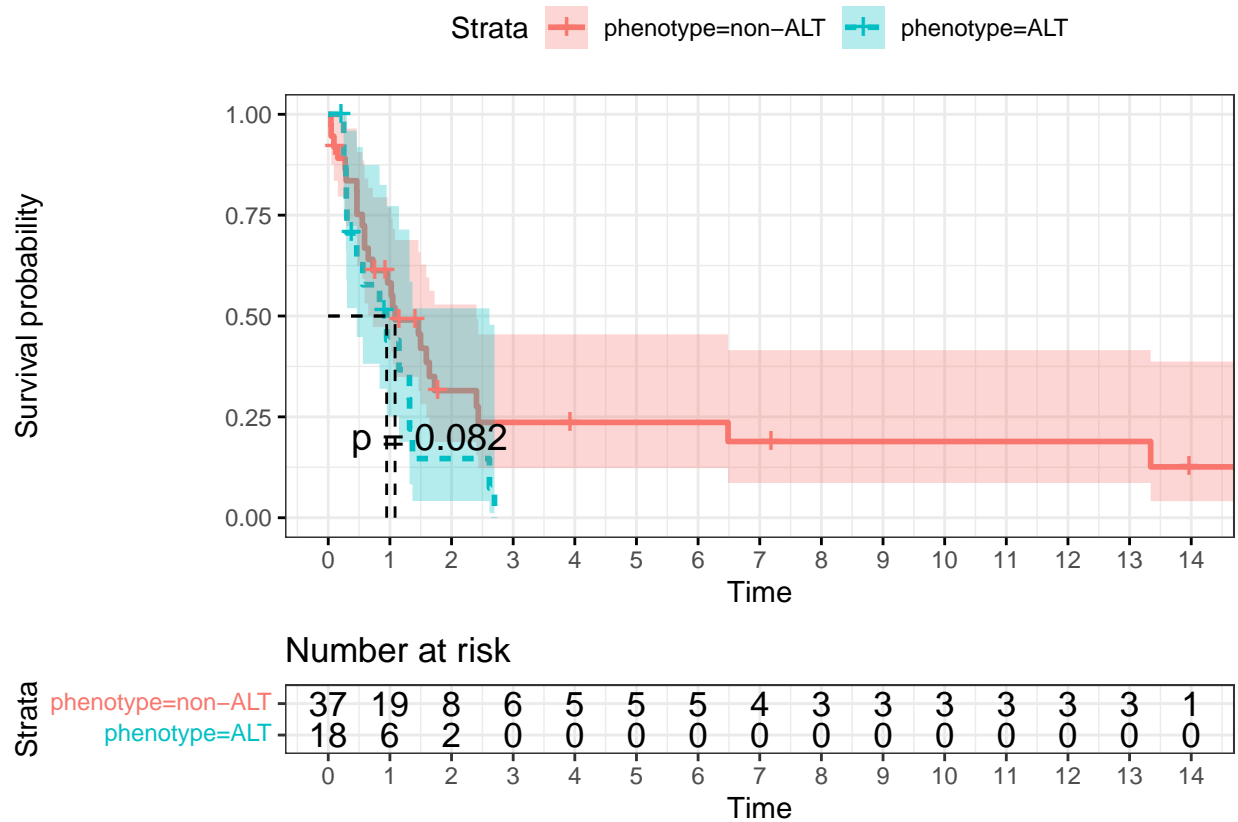
# Save the plot
cowplot::save_plot(filename = file.path(plots_dir,
                                       paste0("logrank_survival_by_", ind_var, "_pfs_only_hgat.png")),
                   plot = surv_plot_logrank)
}

```









Multivariate analysis

```
# define multi-variables that we are using for analyzing survival
list_of_variates <- c("dmg_h3k28+telhunt_cat+reported_gender+atr_x_mut",
                      "dmg_h3k28+telhunt_score+reported_gender+atr_x_mut",
                      "dmg_h3k28+phenotype+reported_gender+atr_x_mut")

# define model
for (ind_var in list_of_variates){
  model <- paste0("survival::Surv(time = PFS_years, event = PFS_status) ~ ", 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, "_pfs_only_hgat")))
```

```

print(table)

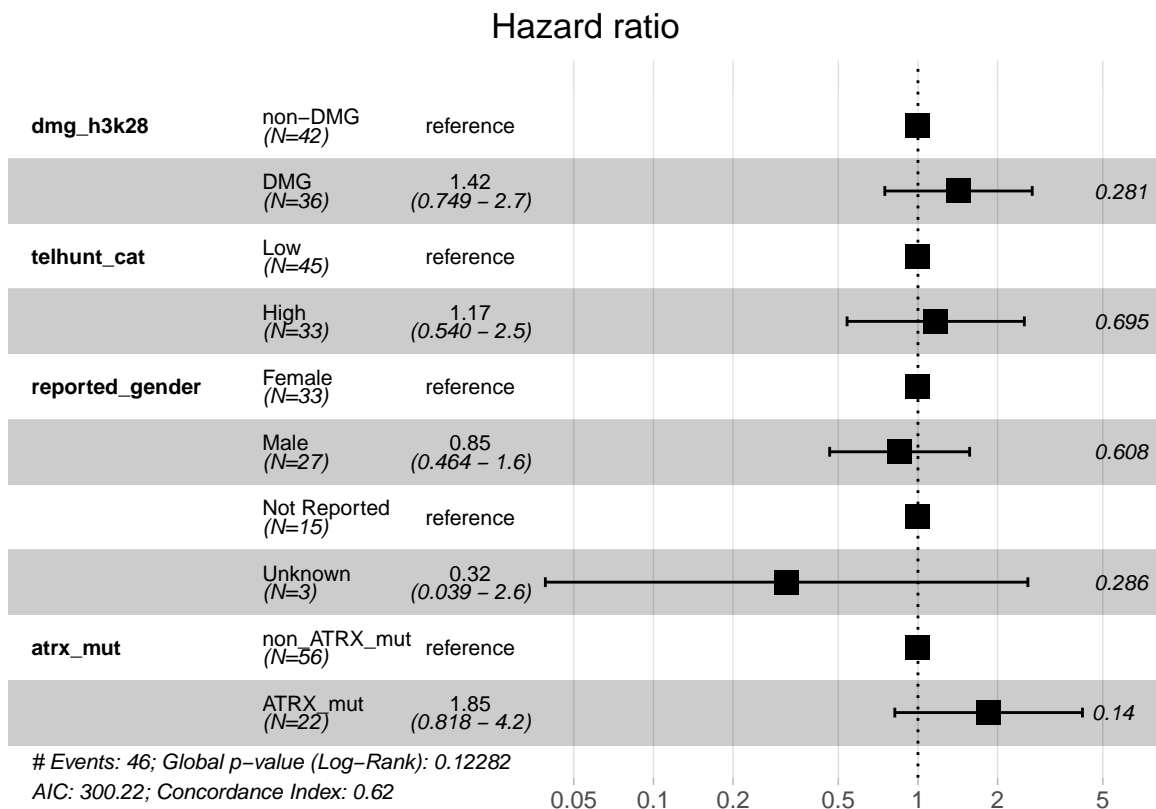
# printout the plot
forest_coxph <- survminer::ggforest(fit, data = data_used)
print(forest_coxph)
}

```

```

## # A tibble: 5 x 5
##   term                estimate std.error statistic p.value
##   <chr>                <dbl>    <dbl>    <dbl>    <dbl>
## 1 dmg_h3k28DMG          0.353      0.327      1.08     0.281
## 2 telhunt_catHigh       0.155      0.394      0.393     0.695
## 3 reported_genderMale  -0.159      0.311     -0.513     0.608
## 4 reported_genderUnknown -1.14       1.07     -1.07     0.286
## 5 atrx_mutATRX_mut      0.615      0.417      1.48     0.140

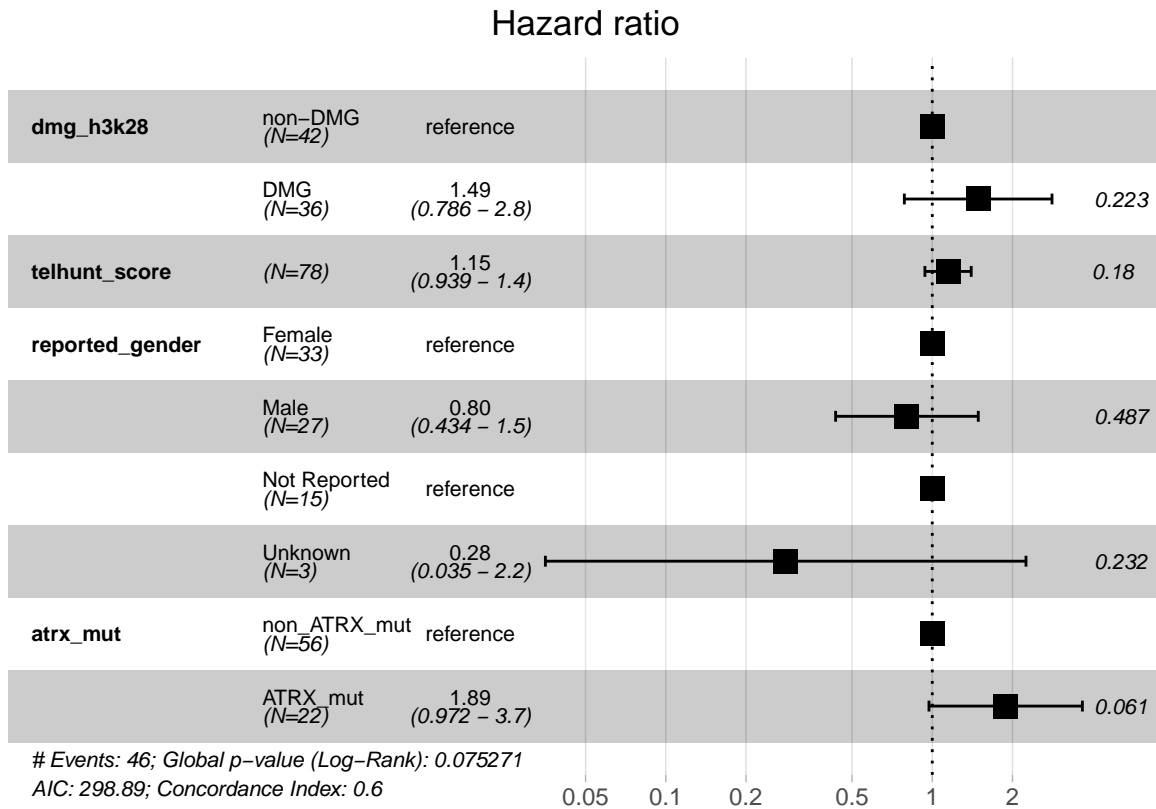
```



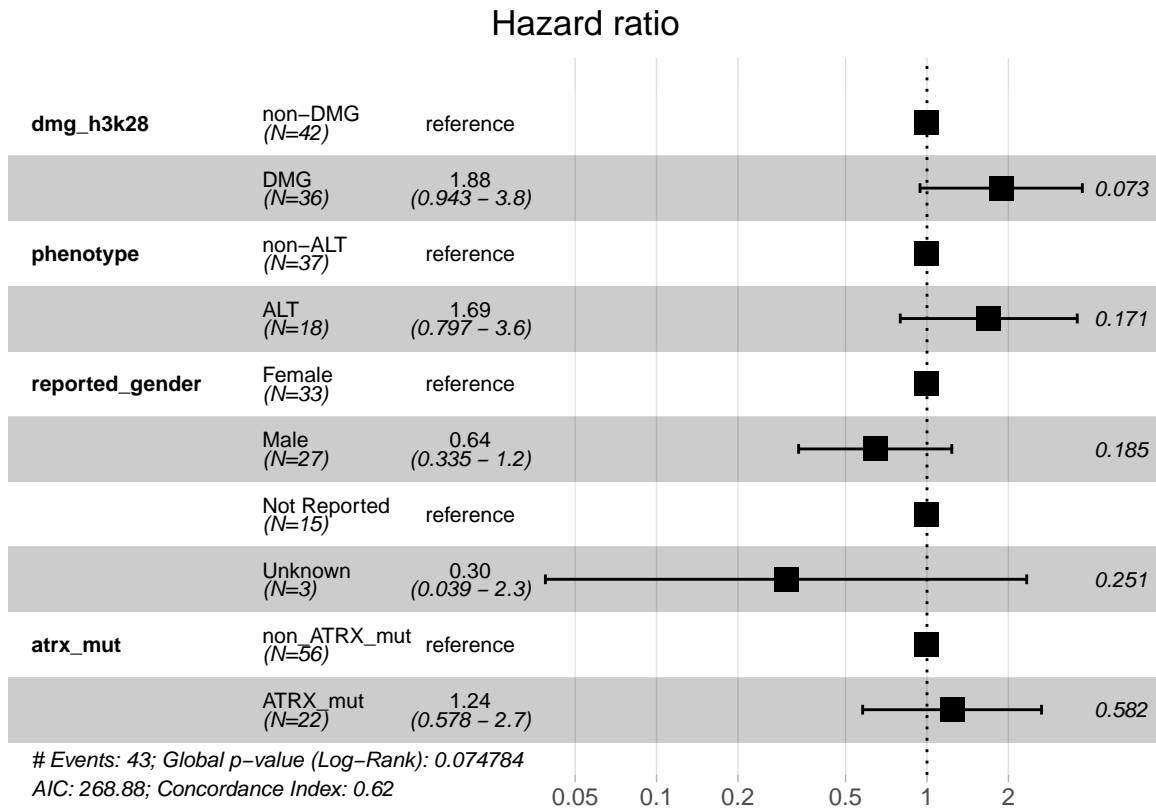
```

## # A tibble: 5 x 5
##   term                estimate std.error statistic p.value
##   <chr>                <dbl>    <dbl>    <dbl>    <dbl>
## 1 dmg_h3k28DMG          0.397      0.325      1.22     0.223
## 2 telhunt_score         0.137      0.102      1.34     0.180
## 3 reported_genderMale  -0.219      0.314     -0.696     0.487
## 4 reported_genderUnknown -1.27       1.06     -1.20     0.232
## 5 atrx_mutATRX_mut      0.635      0.338      1.88     0.0606

```

```
## # A tibble: 5 x 5
##   term                estimate std.error statistic p.value
##   <chr>                <dbl>    <dbl>    <dbl>    <dbl>
## 1 dmgh3k28DMG          0.632    0.353     1.79    0.0731
## 2 phenotypeALT         0.526    0.384     1.37    0.171
## 3 reported_genderMale -0.441    0.333    -1.32    0.185
## 4 reported_genderUnknown -1.20    1.05    -1.15    0.251
## 5 atr_x_mutATRX_mut    0.214    0.389     0.551    0.582
```



Run without ATRX mutants

```
meta_formatted_hgat_non_atrx <- meta_formatted_hgat %>%
  dplyr::filter(atrx_mut == "non_ATRX_mut")

for(ind_var in c("telhunt_cat", "phenotype")){
  # define model
  model <- paste0("survival::Surv(time = PFS_years, event = PFS_status) ~ ", ind_var)

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

  # run survival analysis
  fit <- survival::survdif(formula(model),
    data = data_used)

  # Obtain p value for Chi-Squared stat
  fit$p.value <- pchisq(fit$schisq, df = length(fit$n) - 1, lower = FALSE)

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

  # 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, "_pfs_only_hgat_non_a",
                                       plot = surv_plot_logrank)

}

```

