Experiment 010

Effect of Rapamycin Treatment on Survival to Systemic Infection with *P. rettgeri*

by Barbara Black

Last compiled on 28 January, 2025

Contents


```
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE, results = 'hide', dpi = 300)
```

```
# Initialise a local Git repo
#usethis::use_git()

# Connect local Git repo to GitHub
#usethis::use_github()
```

Background

Code

Functions

```r
# function to change code chunk size in output
# note: automatically applied when chunk runs
def.chunk.hook = knitr::knit_hooks$get("chunk")
knitr::knit_hooks$set(chunk = function(x, options) {
  x = def.chunk.hook(x, options)
  ifelse(options$size != "normalsize", paste0("\n \\", options$size,"\n\n", x, "\n\n \\normalsize"), x)
})

# function to turn $censored into 'character' across all files
# note: some Benching files are exported with $censored as 'double'
read_contents = function(file) {
  df = read_csv(file.path(path, file))
  #df = df %>%
    #mutate(censored = as.character(censored))
  return(df)
}

# function to check and correct miscounted deaths
# note: use with grouped dataset and within apply()
correct_status = function(row) {
  status = row['status']
  n = row['n']
  censored = row['censored']
  status = as.numeric(status)
  n = as.numeric(n)
  censored = as.numeric(censored)
  if (status > n) {
    return(n - censored)
  } else {
    return(status)
  }
}

# function to extract table from mixed effects model (by David Duneau)
extract_coxme_table = function (mod){
    beta = fixef(mod)
    nvar = length(beta)
    nfrail = nrow(mod$var) - nvar
    se = sqrt(diag(mod$var)[nfrail + 1:nvar])
    z = round(beta/se, 2)
    p = format(as.numeric(pchisq((beta/se)^2, 1,lower.tail = FALSE)), 4)
    table = data.frame(cbind(beta,se,z,p))
    return(table)
}

# function to extract and plot survival data (by David Duneau)
ggplotprep2 <- function(x, times){
  # spreading the survfit data frame into data frame per day
  d <- data.frame(condition=rep(names(x$strata), x$strata), time=x$time, survival=x$surv, upper=x$upper
  # function to add time point 0
  fillup0 <- function(s) rbind(c(condition=s, time=0, survival=1, upper=1, lower=1), d[d$condition==s,
  # function to determine the missing time points
  indexes <- function(x, time) {
```

```
    if(x%in%time) return(x)
    return(time[which.min(abs(time[time<x]-x))])
  }
  # function to complete the missing time points
  fillup <- function(s) {
    d.temp <- d[d$condition==s, ]
    time <- as.numeric(d.temp$time)
    id <- sapply(times, indexes, time=time)
    d.temp <- d.temp[match(id, time), ]
    d.temp$time <- times
    return(d.temp)
  }
  if(times[1]==0) d <- do.call("rbind", sapply(names(x$strata), fillup0, simplify=F))
  d <- do.call("rbind", sapply(names(x$strata), fillup, simplify=F))
  d <- data.frame(Condition=d$condition, Time=as.numeric(d$time), Survival=as.numeric(d$survival), Uppe
  return(d)
}

# additional functions (by David Duneau)
grab_grob <- function(){
  grid.echo()
  grid.grab()
}

RIGHT = function(x,n){
  substring(x,nchar(x)-n+1)
}

LEFT = function(x,n){
  substring(x,1,nchar(x)-n+1)
}

r2.corr.mer <- function(m) {
  lmfit <-  lm(model.response(model.frame(m)) ~ fitted(m))
  summary(lmfit)$r.squared
}

logit2prob <- function(logit){
  odds <- exp(logit)
  prob <- odds / (1 + odds)
  return(prob)
}

`%notin%` = Negate(`%in%`)
```

Libraries

```
library(openxlsx)
library(survival)
library(ggsurvfit)
library(car)
```

```
library(gtsummary)
library(survminer)
library(janitor)
library(ggsci)
library(ggtext)
library(mltools)
library(broom)
library(coxme)
library(formatR)
library(tidyverse)
library(cowplot)
library(knitr)
library(stargazer)
```

Themes | `ggplot()`

```
SuperSmallfont= 10
Smallfont= 12
Mediumfont= 14
Largefont= 14
verylargefont = 16
pointsize= 0.7
linesize=0.35
meansize = 1.5
Margin=c(0,0,0,0)

fontsizeaxes = 12
fontsizeaxes2 = 10

basic_theme_surv=
  theme(aspect.ratio = 1,
        panel.background = element_blank(),
        plot.caption =element_text(size=SuperSmallfont,face="italic", hjust = 0.5),
        plot.title = element_text(size=Mediumfont,face="bold", hjust = 0.5),
        plot.subtitle = element_text(size=Smallfont, hjust = 0.5),
        strip.text.x = element_markdown(size =Smallfont, colour = "black",face="italic",hjust = 0.5),
        strip.text.y = element_markdown(size =Smallfont, colour = "black",face="italic",hjust = 0.5),
        strip.background = element_rect(fill=NA, colour="black"),
        strip.placement="outside",
        axis.title.x = element_text(size=Mediumfont,colour="black"),
        axis.title.y = element_text(size=Mediumfont,colour="black"),
        axis.line.x = element_line(colour="black",size=0.75),
        axis.line.y = element_line(colour="black",size=0.75),
        axis.ticks.x = element_line(size = 0.75),
        axis.ticks.y = element_line(size = 0.75),
        axis.text.x = element_text(size=Smallfont,colour="black"),
        axis.text.y = element_text(size=Smallfont,colour="black"),
        plot.margin = unit(Margin, "cm"),
        legend.direction = "vertical",
        legend.box = "vertical",
        legend.position = "right",
```

```
        legend.key.height = unit(0.4, "cm"),
        legend.key.width= unit(0.6, "cm"),
        legend.title = element_text(face="italic",size=Smallfont),
        legend.key = element_rect(colour = 'white', fill = "white", linetype='dashed'),
        legend.text = element_text(size=SuperSmallfont),
        legend.background = element_rect(fill=NA))

colours = palette.colors()
```

Data Import

```
# set path and read data files
path = here::here("data/input")

data_files = list.files(path,
                        pattern = "*.csv")

# standardise all columns as character variables across all data files
data_list = data.frame(filename = data_files) %>%
  mutate(file_contents = map(filename,
                             ~ read_contents(.)))

# unnest the data list into a single data frame
data_raw = unnest(data_list,
                  cols = file_contents)

# print the result
print(data_raw)
```

Data Processing

```
data = data_raw %>%
  # Remove unnecessary variables
  select(-filename) %>%
  # Turn character variables into factor ones, except for `$Day_of_infection` and `$Time`
  mutate(across(.cols = where(is.character) & !all_of(c("Day_of_infection", "Time")),
                .fns = as.factor),
         Date_Time = paste(Day_of_infection, Time, sep = " "),
         Date_Time = as.POSIXct(Date_Time, format = "%d/%m/%Y %H:%M:%S")) %>%
  # Arrange rows by the values of `$date_time`
  arrange(Date_Time) %>%
  # Group dataset by factor variables
  group_by(across(where(is.factor))) %>%
  # Calculate time difference to create `$Time_to_death`
  mutate(Start_time = first(Date_Time),
         across(c(Start_time, Date_Time), ~as.POSIXct(., format = "%d/%m/%Y %H:%M:%S")),
         Time_to_death = difftime(Date_Time, Start_time, units = "hours"),
         Time_to_death = as.numeric(Time_to_death),
         Time_to_death = case_when(Time_to_death > 0 & Time_to_death <= 8 ~ 8,
```

```
                                    #Time_to_death > 2 & Time_to_death <= 4 ~ 4,
                                    #Time_to_death > 4 & Time_to_death <= 8 ~ 8,
                                    Time_to_death > 8 & Time_to_death <= 16 ~ 16,
                                    Time_to_death > 16 & Time_to_death <= 24 ~ 24,
                                    Time_to_death > 24 & Time_to_death <= 32 ~ 32,
                                    Time_to_death > 32 & Time_to_death <= 40 ~ 40,
                                    Time_to_death > 40 & Time_to_death <= 48 ~ 48,
                                    Time_to_death > 48 & Time_to_death <= 56 ~ 56,
                                    TRUE ~ Time_to_death)) %>%
  # Remove unnecessary variables
  select(-c(Start_time, Date_Time)) %>%
  # Generate row IDs
  rowid_to_column()
```

```
# Create dataset with initial sample size for each experimental group
sample_size = data %>%
  filter(Time_to_death == 0) %>%
  group_by(across(where(is.factor))) %>%
  summarise(Sample_size = Count)

# Create dataset with total number of scored flies for each experimental group
scored = data %>%
  filter(Time_to_death > 0) %>%
  group_by(across(where(is.factor))) %>%
  summarise(Events = sum(Count))

# Check for miscounted data by calculating the difference between sample size and total number of score
negative_values = left_join(sample_size, scored) %>%
  mutate(Final_count = Sample_size - Events) %>%
  filter(Final_count < 0)

# Print experimental groups showing negative count values
print(negative_values)

# Export dataset as CSV file
# Note: Dataset is not yet expanded
write_csv(data, "data/output/exp_010_systemic.csv")
```

Survival

```
data_expanded = data %>%
  # Remove original row IDs
  select(-rowid) %>%
  # Expand dataset by duplicating rows according to values from `$Count`
  uncount(Count) %>%
  # Generate new row IDs
  rowid_to_column() %>%
  # Turn $Day_of_infection into factor
  mutate(Day_of_infection = as.factor(Day_of_infection),
         Treatment = fct_relevel(Treatment, rev),
         Age = fct_relevel(Age, rev))
```

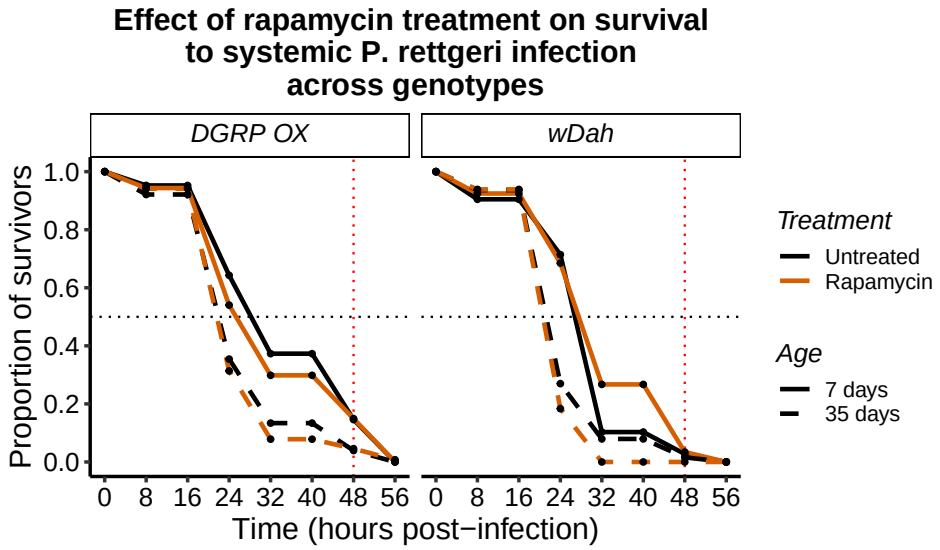
```r
num_observations = data_expanded %>%
  filter(Time_to_death == 0 & Censor == 0) %>%
  group_by(Line) %>%
  summarise(count = n(), .groups = "drop")

plot_caption = num_observations %>%
  mutate(summary = paste(Line, "=", count)) %>%
  pull(summary) %>%
  paste(collapse = "; ")
```

```r
fit_genotype = survfit(Surv(Time_to_death, Censor) ~ Treatment + Age + Line,
                   data = data_expanded)

plot_genotype = ggplotprep2(fit_genotype, times = c(seq(0, 56, by = 8))) %>%
  mutate(Condition = str_remove_all(Condition,"[a-zA-Z]*=")) %>%
  separate(Condition, c("Treatment", "Age", "Line"), sep = ", ") %>%
  mutate(across(where(is.character), as.factor),
         Treatment = fct_relevel(Treatment, rev),
         Age = fct_relevel(Age, rev))

ggplot(plot_genotype, aes(x = Time, y = Survival, colour = Treatment)) +
  facet_grid(~Line) +
  geom_hline(yintercept = 0.5, linetype = "dotted") +
  geom_vline(xintercept = 48, colour = "red", linetype = "dotted") +
  geom_line(aes(linetype = Age), linewidth = 1) +
  geom_point(colour = "black", size = 1) +
  scale_color_manual(values = colours[c(1,7)]) +
  scale_linetype_manual(values = c("solid", "dashed")) +
  labs(title = "Effect of rapamycin treatment on survival \nto systemic P. rettgeri infection \nacross g
       caption = paste("Number of observations (n) per genotype:", plot_caption)) +
  scale_y_continuous("Proportion of survivors",
                     limits = c(0, 1),
                     breaks = c(0, 0.2, 0.4, 0.6, 0.8, 1)) +
  scale_x_continuous("Time (hours post-infection)",
                     breaks = c(seq(0, 56, by = 8))) +
  basic_theme_surv
```



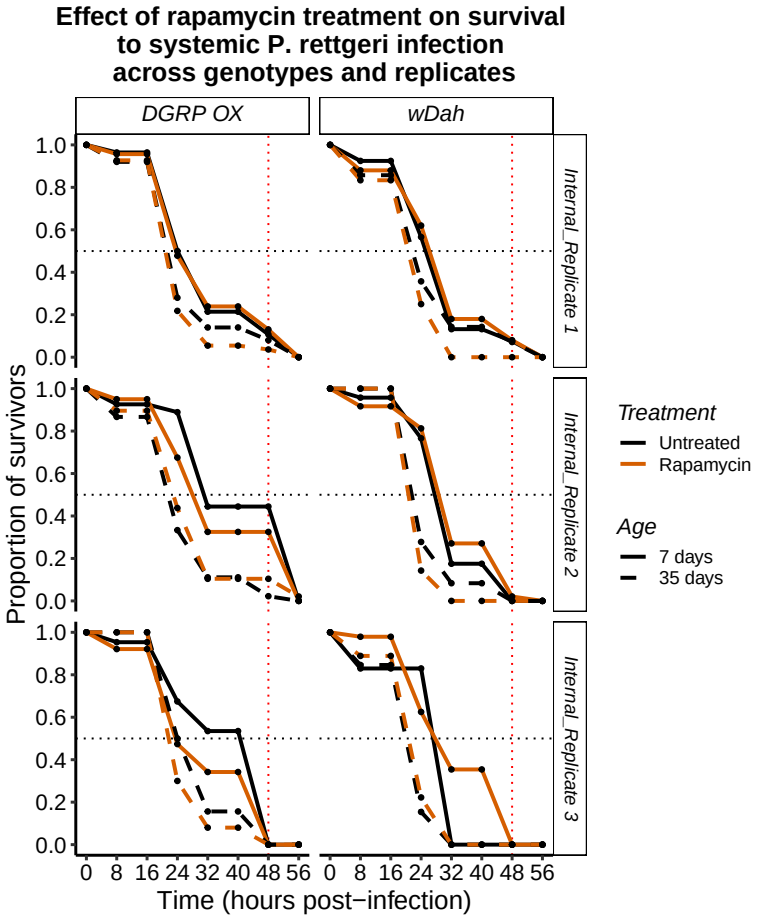
Number of observations (n) per genotype: DGRP OX = 616; wDah = 422

```
fit_replicate = survfit(Surv(Time_to_death, Censor) ~ Treatment + Age + Line + Internal_replicate,
                    data = data_expanded)

plot_replicate = ggplotprep2(fit_replicate, times = c(seq(0, 56, by = 8))) %>%
  mutate(Condition = str_remove_all(Condition,"[a-zA-Z]*=")) %>%
  separate(Condition, c("Treatment", "Age", "Line", "Internal_replicate"), sep = ", ") %>%
  mutate(across(where(is.character), as.factor),
         Treatment = fct_relevel(Treatment, rev),
         Age = fct_relevel(Age, rev))

ggplot(plot_replicate, aes(x = Time, y = Survival, colour = Treatment)) +
  facet_grid(~Internal_replicate~Line) +
  geom_hline(yintercept = 0.5, linetype = "dotted") +
  geom_vline(xintercept = 48, colour = "red", linetype = "dotted") +
  geom_line(aes(linetype = Age), linewidth = 1) +
  geom_point(colour = "black", size = 1) +
  scale_color_manual(values = colours[c(1,7)]) +
  scale_linetype_manual(values = c("solid", "dashed")) +
  labs(title = "Effect of rapamycin treatment on survival \nto systemic P. rettgeri infection \nacross g
       caption = paste("Number of observations (n) per genotype:", plot_caption)) +
  scale_y_continuous("Proportion of survivors",
                     limits = c(0, 1),
                     breaks = c(0, 0.2, 0.4, 0.6, 0.8, 1)) +
```

```
  scale_x_continuous("Time (hours post-infection)",
                     breaks = c(seq(0, 56, by = 8))) +
  basic_theme_surv +
  theme(strip.text.y = element_markdown(size = SuperSmallfont,
                                        colour = "black",
                                        face = "italic",
                                        hjust = 0.5))
```



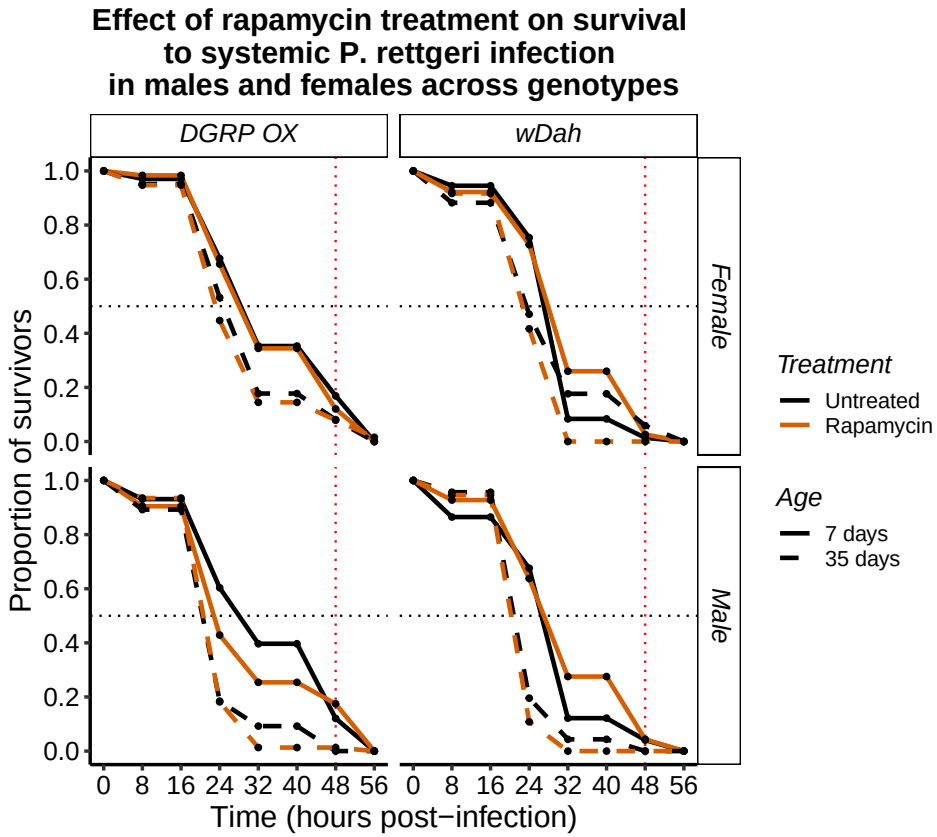
Number of observations (n) per genotype: DGRP OX = 616; wDah = 422

```
fit_sex = survfit(Surv(Time_to_death, Censor) ~ Treatment + Age + Line + Sex,
                  data = data_expanded)

plot_sex = ggplotprep2(fit_sex, times = c(seq(0, 56, by = 8))) %>%
  mutate(Condition = str_remove_all(Condition,"[a-zA-Z]*=")) %>%
  separate(Condition, c("Treatment", "Age", "Line", "Sex"), sep = ", ") %>%
  mutate(across(where(is.character), as.factor),
         Treatment = fct_relevel(Treatment, rev),
         Age = fct_relevel(Age, rev))

ggplot(plot_sex, aes(x = Time, y = Survival, colour = Treatment)) +
  facet_grid(~Sex~Line) +
  geom_hline(yintercept = 0.5, linetype = "dotted") +
  geom_vline(xintercept = 48, colour = "red", linetype = "dotted") +
```

```
  geom_line(aes(linetype = Age), linewidth = 1) +
  geom_point(colour = "black", size = 1) +
  scale_color_manual(values = colours[c(1,7)]) +
  scale_linetype_manual(values = c("solid", "dashed")) +
  labs(title = "Effect of rapamycin treatment on survival \nto systemic P. rettgeri infection \nin males
       caption = paste("Number of observations (n) per genotype:", plot_caption)) +
  scale_y_continuous("Proportion of survivors",
                     limits = c(0, 1),
                     breaks = c(0, 0.2, 0.4, 0.6, 0.8, 1)) +
  scale_x_continuous("Time (hours post-infection)",
                     breaks = c(seq(0, 56, by = 8))) +
  basic_theme_surv
```



Number of observations (n) per genotype: DGRP OX = 616; wDah = 422

Regression

Mixed Genotypes

```
model1 = coxme(Surv(Time_to_death, Censor) ~
                         Age + Treatment + Sex + Line + (1|Internal_replicate) + (1|Vial_ID),
                       data = data_expanded)
```

```
summary(model1)


model2 = coxme(Surv(Time_to_death, Censor) ~
                         Age + Treatment + Sex + Line + Treatment*Sex + (1|Internal_replicate) + (1|Vial
                       data = data_expanded)
summary(model2)
test2 = anova(model1, model2)
print(test2)

model3 = coxme(Surv(Time_to_death, Censor) ~
                         Age + Treatment + Sex + Line + Treatment*Line + (1|Internal_replicate) + (1|Via
                       data = data_expanded)
summary(model3)
test3 = anova(model1, model3)
print(test3)


model4 = coxme(Surv(Time_to_death, Censor) ~
                         Age + Treatment + Sex + Line + Sex*Line + (1|Internal_replicate) + (1|Vial_ID)
                       data = data_expanded)
summary(model4)
test4 = anova(model1, model4)
print(test4)

best_model = model4
```

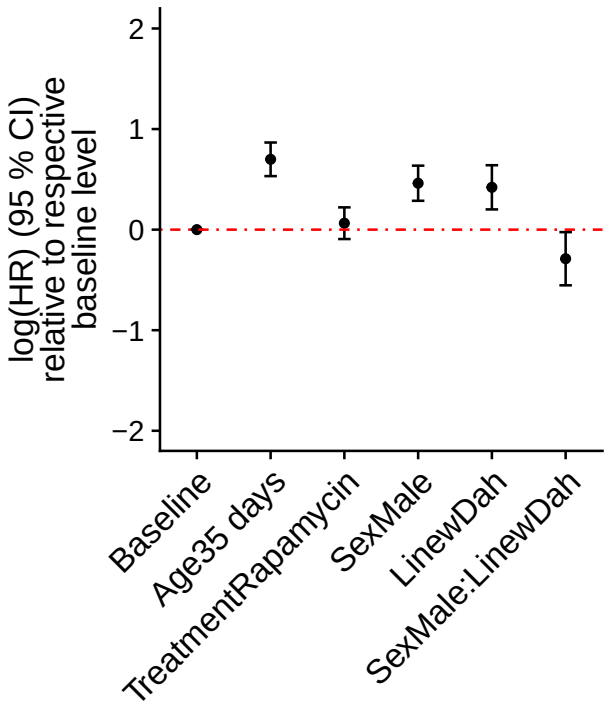
```
coxme_table = extract_coxme_table(best_model)
coxme_table$beta_lower = confint(best_model)[,1]
coxme_table$beta_upper = confint(best_model)[,2]

coxme_table = rownames_to_column(coxme_table, "Term")
coxme_table = coxme_table %>%
  mutate(across(beta:beta_upper, as.numeric))

forest_plot = coxme_table %>%
  add_row(Term = "Baseline",
          beta = 0,
          se = 0,
          z = 0,
          p = 0,
          beta_lower = 0,
          beta_upper = 0) %>%
  mutate(Term = as_factor(Term) %>%
           fct_relevel("Baseline"))

ggplot(forest_plot,
       aes(x = Term,
           y = beta)) +
   geom_errorbar(aes(ymin = beta_lower,
                     ymax = beta_upper),
                 col = "black",
                 width = 0.17,
```

```
                size = 0.5,
                show.legend = FALSE) +
  geom_point(stat = "identity",
             show.legend = FALSE,
             size = 1.5,
             #position = position_dodge(0.2)
             ) +
  geom_hline(yintercept = 0,
             colour = "red",
             linetype = 4) +
  scale_color_manual(values = c("red","blue","green","orange","black")) +
  scale_x_discrete(expand = c(0.1, 0)) +
  scale_y_continuous("log(HR) (95 % CI)<br/>relative to respective<br/>baseline level",
                     breaks = c(seq(-5, 5, by = 1))) +
  coord_cartesian(ylim = c(-2, 2),
                  expand = TRUE,
                  clip = "on") +
  xlab("") +
  theme_cowplot() +
  theme(axis.title.y = ggtext::element_markdown(),
        axis.text.x = element_text(angle = 45,
                                   hjust = 1,
                                   size = 14),
        aspect.ratio = 1)
```




```
library(texreg)
# regression table
coxme_gt = tbl_regression(best_model,
```

```
                          exponentiate = FALSE)
# regression formula
best_formula = formula(best_model)

coxme_gt %>%
  modify_header(update = c(estimate ~ "**log(HR)**",
                           label ~ "**Variable**")) %>%
  italicize_levels() %>%
  modify_caption(caption = paste("$\\", str_replace_all(deparse(best_formula), "_", "\\\\_"),  "$", coll

# paste("$\\", deparse(best_formula), "$", collapse = ""))

stargazer(model4, type = "html", out = "coxme_table.html")

coxme_gt %>%
  as_gt() %>%
  gt::as_latex()
```

Table 1: $ Surv(Time_to_death, Censor) ~ Age + Treatment + Sex + Line + $$ Sex * Line + (1 | Internal_replicate) + (1 | Vial_ID) $

Variable	**log(HR)**	**95% CI**	**p-value**
Age			
7 days	—	—	
35 days	0.70	0.53, 0.87	<0.001
Treatment			
Untreated	—	—	
Rapamycin	0.06	-0.09, 0.22	0.4
Sex			
Female	—	—	
Male	0.46	0.29, 0.64	<0.001
Line			
DGRP OX	—	—	
wDah	0.42	0.20, 0.64	<0.001
Sex * Line			
*Male * wDah*	-0.29	-0.55, -0.02	0.032

```
##
## % Error: Unrecognized object type.
```

Characteristic	**Beta**	**95% CI**[1]	**p-value**
Age			
7 days	—	—	
35 days	0.70	0.53, 0.87	<0.001
Treatment			
Untreated	—	—	
Rapamycin	0.06	-0.09, 0.22	0.4
Sex			

Female	—	—	
Male	0.46	0.29, 0.64	<0.001
Line			
DGRP OX	—	—	
wDah	0.42	0.20, 0.64	<0.001
Sex * Line			
Male * wDah	-0.29	-0.55, -0.02	0.032

[1] CI = Confidence Interval