Figure 1E

Toshihiro Arae

General directory setting

```
wd <- here::here()
shared <- fs::path(fs::path_dir(wd), "shared")
```

Loading packages

```
library(magrittr)
library(ggplot2)
```

Load common R scripts

```
#source(fs::path(wd, "script_r", "MISC.R"))
#source(fs::path(here::here(), "script_r", "MISC_PALETTE.R"))
```

Load script

```
source(fs::path(wd, "script_r", "MISC_FIG.R"))
readLines(fs::path(wd, "script_r", "MISC_FIG.R")) %>% cat(sep = "\n")
```

```
library(magrittr)
library(ggplot2)

COL_PALETTE <-
  viridis::inferno(6, begin = .1, end = .9) %>%
  rev() %>%
  setNames(nm = c("ZT0", "ZT3", "ZT6", "ZT12", "ZT18", "ZT21"))

LABEL_PALETTE <-
  COL_PALETTE %>%
  prismatic::clr_darken(shift = .15) %>%
  setNames(names(COL_PALETTE))

label_number_si <-
  purrr::partial(scales::label_number, scale_cut = scales::cut_short_scale())

ggsave_single <- function(..., width = 86, height = 230, dpi = 300) {
  f <- purrr::partial(ggsave, width = width, height = height, dpi = dpi, units = "mm")
  f(...)
}

ggsave_double <- function(..., width = 178, height = 230, dpi = 300) {
  f <- purrr::partial(ggsave, width = width, height = height, dpi = dpi, units = "mm")
  f(...)
}

#' Utility functions for making secondary y-axis
#' @param y1 numeric vector
#' @param y2 numeric vector
#' @name util_2nd_axis
```

```
#' @examples
#' make_scale_y1_to_y2(1:5, 6:10)(1:10)
#' make_scale_y2_to_y1(1:5, 6:10)(1:10)
#'
#' iris_ <- dplyr::select(iris, x = Sepal.Length, y1 = Petal.Length, y2 = Petal.Width)
#' gp1 <-
#'   iris_ %>%
#'   ggplot() +
#'   geom_point(aes(x, y1), color = "#CD3700") +
#'   geom_point(aes(x, y2), color = "#473C8B")
#'
#' to_y1 <- with(iris_, {make_scale_y2_to_y1(y1, y2)})
#' to_y2 <- with(iris_, {make_scale_y1_to_y2(y1, y2)})
#' gp2 <-
#'   iris_ %>%
#'   ggplot() +
#'   geom_point(aes(x, y1), color = "#CD3700") +
#'   geom_point(aes(x, y = to_y1(y2)), color = "#473C8B") +
#'   scale_y_continuous(sec.axis = sec_axis(trans = to_y2, name = "y2"))
#' patchwork::wrap_plots(gp1, gp2)
#'
NULL

#' Create transformation function of range(y1) to range(y2)
#' @rdname util_2nd_axis
#' @export
#'
make_scale_y1_to_y2 <- function(y1, y2) {
  function(n) {
    scales:::rescale.numeric(
      n,
      to = range(y2, na.rm = TRUE, finite = TRUE),
      from = range(y1, na.rm = TRUE, finite = TRUE)
    )
  }
}

#' Create transformation function of range(y2) to range(y1)
#' @rdname util_2nd_axis
#' @export
#'
make_scale_y2_to_y1 <- function(y1, y2) {
  function(n) {
    scales:::rescale.numeric(
      n,
      to = range(y1, na.rm = TRUE, finite = TRUE),
      from = range(y2, na.rm = TRUE, finite = TRUE)
    )
  }
}

#' Create transformation function of range(y2) to range(y1)
#' @rdname util_2nd_axis
#' @export
#'
make_scale_y2_to_y1_se <- function(y1, y2) {
  to <- range(y1, na.rm = TRUE, finite = TRUE)
  from <- range(y2, na.rm = TRUE, finite = TRUE)
  function(n) n / (diff(from) / diff(to))
}
```

Directory setting

```
dir_output <- fs::path("analysis", "fig", "fig01E")
path_out <- function(...) fs::path(wd, dir_output, ...)
fs::dir_create(path_out())
```

Load input data

```
inf_rna <- fs::path(wd, "data_preproc", "readcount",
                "count_rna_exon", "count_by_gene.csv")
inf_ccds <- fs::path(wd, "data_preproc", "readcount",
                "count_ribo_central_cds_psite", "count_by_gene.csv")

tbl_tpm <-
  dplyr::inner_join(
    inf_rna %>%
      readr::read_csv() %>%
      dplyr::select(AGI = Geneid, dplyr::matches("tpm")),
    inf_ccds %>%
      readr::read_csv() %>%
      dplyr::select(AGI = Geneid, dplyr::matches("tpm")),
    by = "AGI"
  )
```

```
Rows: 36917 Columns: 26
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (1): Geneid
dbl (25): Length, zt0_1_rna, zt0_2_rna, zt12_1_rna, zt12_2_rna, zt18_1_rna, ...

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.
Rows: 27631 Columns: 26
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (1): Geneid
dbl (25): Length, zt0_1_ribo, zt0_2_ribo, zt12_1_ribo, zt12_2_ribo, zt18_1_r...

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.
```

```
sample_name <-
  colnames(tbl_tpm)[-1] %>%
  stringr::str_remove("^tpm_") %>%
  stringr::str_remove("_psite$") %>%
  stringr::str_to_upper() %>%
  stringr::str_replace("RIBO", "Ribo")
```

Plotting

Data preparation

```
# Filtering by TPM and variance
# 1. Extract genes with TPM > 3 in more than 12 samples.
# 2. Extract top 3000 genes with high variance from step 1.
tbl_data <-
  tbl_tpm %>%
  {dplyr::filter(., rowSums(dplyr::across(!AGI, ~ . > 3)) >= 12)} %>%
  dplyr::mutate(
    .by = AGI,
```

```
    var =
      unlist(dplyr::c_across(dplyr::starts_with("tpm"))) %>%
      {mean(var(.[1:12]), var(.[13:24]))}
  ) %>%
  dplyr::arrange(desc(var)) %>%
  dplyr::slice_head(n = 3000) %>%
  dplyr::select(!var)

# Sort samples by hiearchical clustering
d <-
  tbl_data %>%
  dplyr::select(-AGI) %>%
  purrr::set_names(sample_name) %>%
  log() %>%
  as.matrix() %>%
  t() %>%
  amap::Dist(method = "correlation")

hc <- hclust(d, method = "average")
dend <- as.dendrogram(hc)

# Z-normalized
tbl_plot <-
  tbl_data %>%
  tidyr::pivot_longer(!AGI) %>%
  dplyr::mutate(
    zt = stringr::str_extract(name, "\\d+") %>% as.integer(),
    type =
      stringr::str_extract(name, "rna|ribo") %>%
      stringr::str_to_upper() %>%
      stringr::str_replace("RIBO", "Ribo") %>%
      forcats::fct_relevel(c("RNA", "Ribo"))
  ) %>%
  dplyr::arrange(AGI, type, zt) %>%
  dplyr::mutate(name = forcats::fct_inorder(name)) %>%
  dplyr::with_groups(c(AGI, type), ~ dplyr::mutate(., scaled = scale(value)[,1]))

# Sort genes by hiearchical clustering
d2 <-
  tbl_plot %>%
  dplyr::select(AGI, name, scaled) %>%
  tidyr::pivot_wider(names_from = name, values_from = scaled) %>%
  tibble::column_to_rownames("AGI") %>%
  amap::Dist(method = "correlation")
hc2 <- hclust(d2, method = "average")
hc2 <- dendsort::dendsort(hc2)
```

Plot dendrogram

```
plot_dendrogram <- function(hc, show_label = FALSE) {
  dend <- as.dendrogram(hc)
  ddata <- ggdendro::dendro_data(dend)
  gp <-
    ggplot() +
    geom_segment(
      data = ggdendro::segment(ddata),
      aes(x = x, y = y, xend = xend, yend = yend),
      linejoin = "round", lineend = "round")

  if(show_label) {
    gp <-
      gp +
```

```
      geom_text(
        data = ggdendro::label(ddata),
        aes(x = x, y = y, label = label),
        angle = 90
      )
  }

  gp <-
    gp +
    scale_x_continuous(expand = expansion(add = 0.5)) +
    scale_y_continuous(expand = expansion(c(0, .1))) +
    theme(
      text = element_blank(),
      line = element_blank(),
      axis.ticks = element_blank(),
      panel.background = element_blank(),
      plot.margin = margin(b = 0)
    )
  gp
}

hc_ <- dendextend::rotate(hc, order = c(13:24, 3:6,1:2,9:12,7:8))
gp1 <- plot_dendrogram(hc_)
```

Plot Heatmap

```
label_order <-
  hc_$labels[hc_$order] %>%
  stringr::str_match_all("[^_]+") %>%
  purrr::map_chr(~ paste0(.x[2], "_*_", .x[1], "_*_", .x[3], "-seq"))

gp2 <-
  tbl_plot %>%
  dplyr::mutate(
    rep = stringr::str_extract(name, "_[12]") %>% stringr::str_sub(start = 2),
    sample_label = paste0(rep, "_*_", "ZT", zt, "_*_", type, "-seq")
  ) %>%
  dplyr::mutate(
    AGI = forcats::fct_relevel(AGI, hc2$labels[hc2$order]),
    sample_label = forcats::fct_relevel(sample_label, label_order)
  ) %>%
  ggplot(aes(sample_label, AGI)) +
  geom_raster(aes(fill = scaled)) +
  geom_vline(xintercept = seq(.5, 23.5, by = 1), color = "white") +
  scale_fill_gradient2(low = scales::muted("blue"), high = scales::muted("red")) +
  scale_x_discrete(
    guide = ggh4x::guide_axis_nested(delim = "_*_"), expand = expansion(add = .5)
  ) +
  guides(fill = guide_colorbar(
    title = "Z-score of TPM",
    title.position = "right",
    barwidth = 2, barheight = 40, default.unit = "mm",
    label.position = "left", label.hjust = 1
  )) +
  theme(
    plot.margin = margin(t = 0),
    legend.title = element_text(angle = -90),
    axis.text.x = element_blank(),
    axis.text.y = element_blank(),
    axis.title = element_blank(),
    axis.ticks = element_blank(),
    ggh4x.axis.nestline.x = element_blank(),
```

```
    ggh4x.axis.nesttext.x = element_blank()
  )
```

Plot sample labels

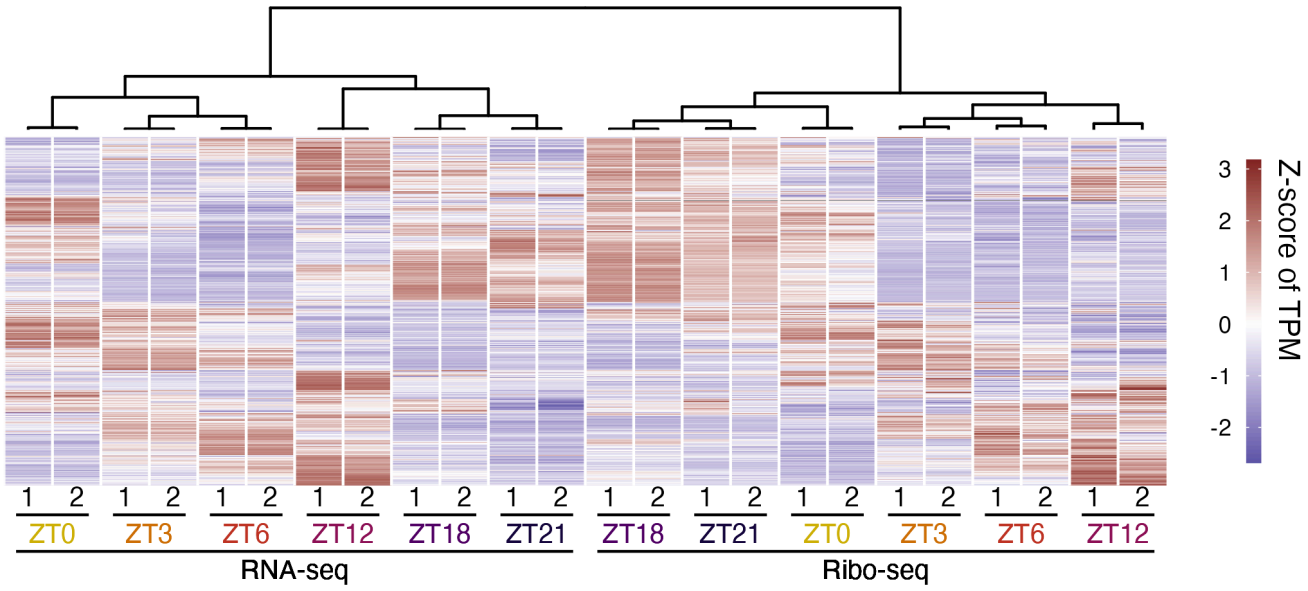
```
tbl_axis <-
  tibble::tibble(label_order) %>%
  dplyr::mutate(
    x = dplyr::row_number(),
    rep = stringr::str_extract(label_order, "^."),
    zt = stringr::str_extract(label_order, "ZT\\d+"),
    type = stringr::str_extract(label_order, "(RNA|Ribo)-seq")
  ) %>%
  dplyr::with_groups(c(type, zt), ~ dplyr::mutate(., x_zt = mean(x))) %>%
  dplyr::with_groups(c(type), ~ dplyr::mutate(., x_type = mean(x)))

ex_bar <- .25
text_size <- 10 / .pt
gp3 <-
  ggplot() +
  geom_text(
    data = tbl_axis,
    aes(x, 3, label = rep),
    size = text_size
  ) +
  geom_text(
    data = dplyr::select(tbl_axis, x_zt, zt),
    aes(x_zt, 2, label = zt, color = zt),
    size = text_size
  ) +
  geom_text(
    data = dplyr::select(tbl_axis, x_type, type),
    aes(x_type, 1, label = type),
    size = text_size
  ) +
  geom_segment(
    data =
      tbl_axis %>%
      dplyr::with_groups(
        .groups = c(type, zt),
        .f = ~ dplyr::summarise(., x_min = min(x) - ex_bar, x_max = max(x) + ex_bar)
      ),
    aes(x = x_min, xend = x_max, y = 2.5, yend = 2.5)
  ) +
  geom_segment(
    data =
      tbl_axis %>%
      dplyr::with_groups(
        .groups = type,
        .f = ~ dplyr::summarise(., x_min = min(x) - ex_bar, x_max = max(x) + ex_bar)
      ),
    aes(x = x_min, xend = x_max, y = 1.5, yend = 1.5)
  ) +
  scale_x_continuous(expand = expansion(add = .5 - ex_bar)) +
  scale_color_manual(
    values =
      COL_PALETTE %>%
      prismatic::clr_darken(shift = .15) %>%
      setNames(names(COL_PALETTE))
  ) +
  theme_void() +
  theme(
```

```
    plot.margin = margin(t = 0, b = 0)
  ) +
  guides(color = guide_none()) +
  coord_cartesian(clip = "off")
```

```
`summarise()` has grouped output by 'type'. You can override using the
`.groups` argument.
```

```
pgp <- patchwork::wrap_plots(gp1, gp2, gp3, ncol = 1, heights = c(.25, .65, .15))
ggsave_double(path_out("fig01G_type6_d.png"), pgp, height = 80)
ggsave_double(path_out("fig01G_type6_d.svg"), pgp, height = 80)
```

```
knitr::include_graphics(path_out("fig01G_type6_d.png"))
```



Sessioninfo

```
sessionInfo()
```

```
R version 4.2.1 (2022-06-23)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Ventura 13.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices datasets  utils     methods   base

other attached packages:
[1] ggplot2_3.4.2  magrittr_2.0.3

loaded via a namespace (and not attached):
 [1] ggh4x_0.2.3         tidyselect_1.2.0    xfun_0.40
 [4] purrr_1.0.1         colorspace_2.0-3    vctrs_0.6.1
```

```
 [7] generics_0.1.3      amap_0.8-19         htmltools_0.5.3
[10] viridisLite_0.4.1   yaml_2.3.6          utf8_1.2.2
[13] rlang_1.1.0         pillar_1.9.0        glue_1.6.2
[16] withr_2.5.0         bit64_4.0.5         lifecycle_1.0.3
[19] stringr_1.5.0       munsell_0.5.0       gtable_0.3.1
[22] ragg_1.2.5          evaluate_0.20       labeling_0.4.2
[25] knitr_1.42          forcats_1.0.0       tzdb_0.3.0
[28] fastmap_1.1.0       parallel_4.2.1      fansi_1.0.3
[31] readr_2.1.4         renv_1.0.3          scales_1.2.1
[34] dendsort_0.3.4      BiocManager_1.30.18 vroom_1.6.0
[37] jsonlite_1.8.4      systemfonts_1.0.4   farver_2.1.1
[40] fs_1.5.2            bit_4.0.5           textshaping_0.3.6
[43] gridExtra_2.3       png_0.1-7           hms_1.1.3
[46] digest_0.6.31       stringi_1.7.12      dplyr_1.1.1
[49] grid_4.2.1          rprojroot_2.0.3     here_1.0.1
[52] cli_3.6.0           tools_4.2.1         patchwork_1.1.2
[55] tibble_3.2.1        ggdendro_0.1.23     dendextend_1.16.0
[58] crayon_1.5.2        tidyr_1.3.0         pkgconfig_2.0.3
[61] MASS_7.3-57         svglite_2.1.0       rmarkdown_2.24
[64] rstudioapi_0.14     viridis_0.6.2       R6_2.5.1
[67] prismatic_1.1.1     compiler_4.2.1
```