

## Figure 2B

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))
}

```

```
dir_output <- fs::path("analysis", "fig", "fig02B")
path_out <- function(...) fs::path(wd, dir_output, ...)
fs::dir_create(path_out())
```

## Define some functions

### Data scaling

```
z_score <- function(x) scale(x)[,1]
centering <- function(x) scale(x, scale = FALSE)[,1]
do_nothing <- function(x) x

centering_then_z_score <- function(x) {
  y1 <- scale(x[1:6], scale = FALSE)[,1]
  y2 <- scale(x[7:12], scale = FALSE)[,1]
  scale(c(y1, y2))[,1]
}
# centering_then_z_score(c(1:6, (7:12)*2)) %>% plot()
```

### Preparing data.frame

```
#' Convert input filtered tibble to tibble for plotting
#' @param tbl a input tibble
#' @param ... will be passed to `dplyr::arrange()`
convert_to_tbl_plot <- function(tbl, ...) {
  tbl <-
    tbl %>%
    dplyr::mutate(rna_coef_zt0 = 0, .before = rna_coef_zt3) %>%
    dplyr::mutate(morf_coef_zt0 = 0, .before = morf_coef_zt3)

  tbl %>%
    dplyr::arrange(...) %>%
    dplyr::mutate(AGI = forcats::fct_inorder(AGI)) %>%
    tidyr::pivot_longer(cols = !c(AGI, rna_Phase, morf_Phase, is_same_peak)) %>%
    dplyr::mutate(
      time =
        stringr::str_extract(name, "zt\\d+") %>%
        stringr::str_to_upper() %>%
        forcats::fct_inorder(),
      type =
        stringr::str_extract(name, "rna|morf") %>%
        {dplyr::case_when(
          . == "rna" ~ "RNA-seq",
          . == "morf" ~ "Ribo-seq"
        )} %>%
        forcats::fct_inorder()
    ) %>%
    dplyr::select(!name) %>%
    tidyr::pivot_wider(names_from = type, values_from = value) %>%
    tidyr::pivot_longer(cols = c("RNA-seq", "Ribo-seq"), names_to = "type") %>%
    dplyr::mutate(type = forcats::fct_relevel(type, c("RNA-seq", "Ribo-seq")))
}

#' Scale the value column of tbl_plot by a given function
#' @param tbl tbl_plot
#' @param scaling_func a function to scale `value`
scale_tbl_plot <- function(tbl, scaling_func) {
  tbl %>%
    dplyr::group_by(AGI) %>%
    dplyr::arrange(type, time) %>%
    dplyr::mutate(scaled = scaling_func(value)) %>%
```

```

    dplyr::ungroup()
  }

sort_tbl <- function(tbl, ...) {
  tbl %>%
    split(.$AGI) %>%
    purrr::map(function(df) {
      temp_time <- paste0("ZT", df$rna_Phase[1])
      if(temp_time == "ZT9") temp_time <- c("ZT6", "ZT12")
      else if(temp_time == "ZT15") temp_time <- c("ZT12", "ZT18")
      peak_rna <-
        df %>%
        dplyr::filter(type == "RNA-seq", time %in% temp_time) %>%
        dplyr::pull(scaled) %>%
        max()
      stopifnot(!is.infinite(peak_rna))

      temp_time <- paste0("ZT", df$morf_Phase[1])
      if(temp_time == "ZT9") temp_time <- c("ZT6", "ZT12")
      else if(temp_time == "ZT15") temp_time <- c("ZT12", "ZT18")
      peak_morf <-
        df %>%
        dplyr::filter(type == "Ribo-seq", time %in% temp_time) %>%
        dplyr::pull(scaled) %>%
        max()
      stopifnot(!is.infinite(peak_morf))

      dplyr::mutate(df, peak_rna, peak_morf)
    }) %>%
    dplyr::bind_rows() %>%
    dplyr::with_groups(c(AGI, rna_Phase, morf_Phase, peak_rna, peak_morf),
      tidyr::nest) %>%
    dplyr::arrange(...) %>%
    tidyr::unnest(cols = data)
  }
}

```

## Heatmap

```

plot_heatmap <- function(tbl) {
  lab <- rev(levels(forcats::fct_inorder(tbl$AGI)))
  tbl_plot <-
    tbl %>%
    dplyr::filter(AGI %in% lab) %>%
    dplyr::mutate(AGI = forcats::fct_relevel(AGI, lab))

  tbl_plot %>%
    ggplot(aes(time, AGI)) +
    geom_tile(aes(fill = scaled)) +
    scale_x_discrete(expand = expansion(0)) +
    scale_y_discrete(expand = expansion(0)) +
    scale_fill_gradient2(low = scales::muted("blue"), high = scales::muted("red")) +
    facet_grid(cols = vars(type)) +
    theme_void(base_size = 10) +
    theme(
      axis.text.x = element_text(angle = 90, color = LABEL_PALETTE),
      strip.text = element_text(size = 10, vjust = 1.1),
      legend.title = element_text(angle = -90),
      plot.margin = margin(5, 5, 5, 5)
    ) +
    guides(fill = guide_colorbar(
      title = "Z-score of Log2FC",
      title.position = "right",

```

```

    barwidth = 2, barheight = 40, default.unit = "mm",
    label.position = "left", label.hjust = 1
  ))
}

theme_empty <- function() {
  list(
    theme_void(),
    theme(
      strip.background = element_blank(),
      strip.text = element_blank(),
      legend.position = "none"
    )
  )
}

```

## Load input data

```

inf <- fs::path(wd, "analysis", "list_summary", "summary_all.csv")
tbl_input <-
  readr::read_csv(inf, show_col_types = FALSE) %>%
  dplyr::filter(!grepl("AT[CM]G", AGI))

```

## Plot

### RNA only

```

tbl_rnaonly <-
  tbl_input %>%
  dplyr::filter(te_morf_padj < 0.05,
    rna_BF_BH < 0.05, (rna_Max_Amp/rna_Max) > 0.5,
    morf_BF_BH >= 0.05, (morf_Max_Amp/morf_Max) <= 0.5) %>%
  dplyr::mutate(is_same_peak = rna_Phase == morf_Phase) %>%
  dplyr::select(AGI, rna_Phase, morf_Phase, is_same_peak,
    dplyr::matches("rna_coef_z"), dplyr::matches("^morf_coef_z"))
tbl_rnaonly %>% dplyr::glimpse()

```

```

Rows: 299
Columns: 14
$ AGI          <chr> "AT1G01830", "AT1G03300", "AT1G10210", "AT1G11410", "AT...
$ rna_Phase    <dbl> 0, 18, 12, 0, 18, 3, 0, 21, 12, 18, 0, 0, 6, 21, 6, 21,...
$ morf_Phase   <dbl> 3, 15, 0, 3, 15, 21, 21, 21, 3, 21, 18, 18, 3, 6, 6, 12...
$ is_same_peak <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, ...
$ rna_coef_zt3 <dbl> -0.60004463, 0.09879740, 0.37084121, 0.02496678, 0.5649...
$ rna_coef_zt6 <dbl> -0.74802693, 0.11332949, 0.67398059, -0.17371054, 0.517...
$ rna_coef_zt12 <dbl> -0.81941981, 0.74172560, 1.17623421, -0.58110416, 1.570...
$ rna_coef_zt18 <dbl> -1.0050460, 0.9259220, 1.0387711, -0.8897882, 1.3754618...
$ rna_coef_zt21 <dbl> -1.0121648, 0.5947502, 0.7408388, -0.8912115, 1.1201118...
$ morf_coef_zt3 <dbl> 0.12795381, -0.40214100, -0.48168731, 0.34545438, -0.39...
$ morf_coef_zt6 <dbl> 0.071574530, -0.138410650, -0.365550140, 0.234717640, -...
$ morf_coef_zt12 <dbl> 0.07752487, -0.69349111, -0.39791378, 0.23263736, 0.075...
$ morf_coef_zt18 <dbl> -0.090049775, -0.056504112, -0.101011797, -0.079566275,...
$ morf_coef_zt21 <dbl> -0.001770316, -0.022370011, -0.201175354, 0.013178462, ...

```

```

tbl_plot <-
  convert_to_tbl_plot(tbl_rnaonly, AGI) %>%
  scale_tbl_plot(scaling_func = centering_then_z_score) %>%

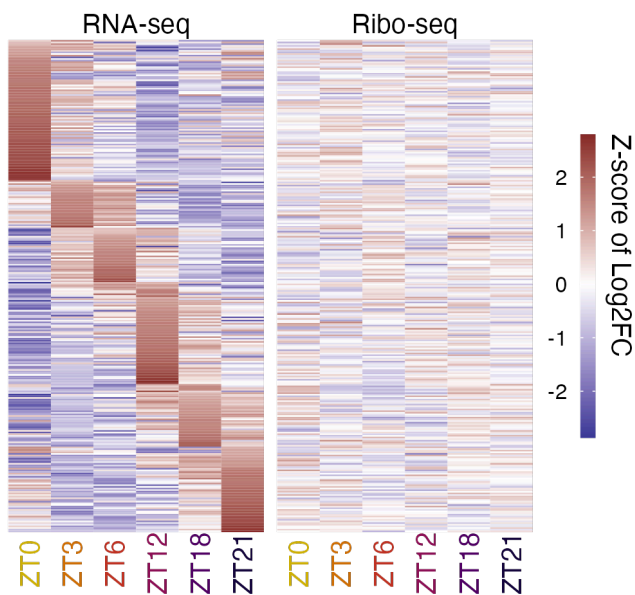
```

```
sort_tbl(rna_Phase, peak_rna)
gp_hm <- plot_heatmap(tbl_plot)
```

Warning: Vectorized input to `element\_text()` is not officially supported.  
 i Results may be unexpected or may change in future versions of ggplot2.

```
ggsave_single(gp_hm, filename = path_out("rna_only_v3.png"), height = 80)
ggsave_single(gp_hm, filename = path_out("rna_only_v3.svg"), height = 80)
ggsave_single(gp_hm + theme_empty(), filename = path_out("rna_only_trim_v3.png"),
              width = 69, height = 65)
```

```
knitr::include_graphics(path_out("rna_only_v3.png"))
```



## Ribo only

```
tbl_riboonly <-
  tbl_input %>%
  dplyr::filter(te_morf_padj < 0.05,
                morf_BF_BH < 0.05, (morf_Max_Amp/morf_Max) > 0.5,
                rna_BF_BH >= 0.05, (rna_Max_Amp/rna_Max) <= 0.5) %>%
  dplyr::mutate(is_same_peak = rna_Phase == morf_Phase) %>%
  dplyr::select(AGI, rna_Phase, morf_Phase, is_same_peak,
                dplyr::matches("rna_coef_z"), dplyr::matches("^morf_coef_z"))
tbl_riboonly %>% dplyr::glimpse()
```

```
Rows: 977
Columns: 14
$ AGI          <chr> "AT1G01110", "AT1G01140", "AT1G01170", "AT1G01760", "AT...
$ rna_Phase    <dbl> 3, 12, 21, 21, 3, 21, 9, 15, 18, 0, 18, 3, 18, 0, 12, 2...
$ morf_Phase   <dbl> 0, 0, 0, 18, 3, 18, 21, 12, 12, 3, 0, 0, 21, 3, 0, 3, 2...
$ is_same_peak <lgl> FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, ...
$ rna_coef_zt3 <dbl> 0.410199030, 0.070894901, 0.238839784, 0.110338635, 0.0...
$ rna_coef_zt6 <dbl> 0.192026704, -0.218919644, 0.234640405, 0.027859305, -0...
$ rna_coef_zt12 <dbl> 0.204927848, 0.601265198, 0.390597761, 0.208967971, -0...
$ rna_coef_zt18 <dbl> 0.327491901, -0.138388197, 0.377208332, 0.641151703, -0...
$ rna_coef_zt21 <dbl> 0.08861642, -0.06221163, 0.37676224, 0.64472237, -0.140...
$ morf_coef_zt3 <dbl> -0.8113399, -0.1837910, -0.2530968, -1.0531747, 0.61792...
$ morf_coef_zt6 <dbl> -1.0230824, -0.9558949, -0.5925671, -1.4295596, 0.58775...
$ morf_coef_zt12 <dbl> -0.62755171, -1.03583078, -0.74238624, -1.84666545, 0.3...
```

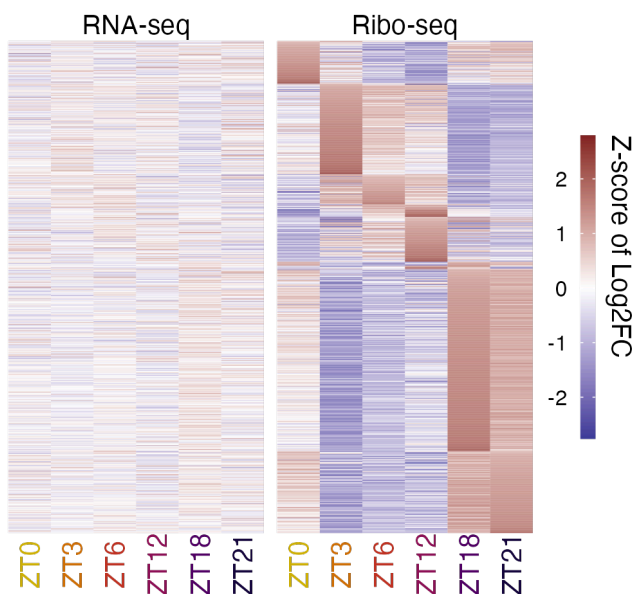
```
$ morf_coef_zt18 <dbl> -0.5981657, -0.6686431, -0.1391774, 0.1568936, -0.23916...
$ morf_coef_zt21 <dbl> -0.46397899, -0.69870615, -0.17879307, 0.07872772, -0.1...
```

```
tbl_plot <-
  convert_to_tbl_plot(tbl_riboonly, AGI) %>%
  scale_tbl_plot(scaling_func = centering_then_z_score) %>%
  sort_tbl(morf_Phase, peak_morf)
gp_hm <- plot_heatmap(tbl_plot)
```

Warning: Vectorized input to `element\_text()` is not officially supported.  
 i Results may be unexpected or may change in future versions of ggplot2.

```
ggsave_single(gp_hm, filename = path_out("ribo_only_v3.png"), height = 80)
ggsave_single(gp_hm, filename = path_out("ribo_only_v3.svg"), height = 80)
ggsave_single(gp_hm + theme_empty(), filename = path_out("ribo_only_trim_v3.png"),
  width = 69, height = 65)
```

```
knitr::include_graphics(path_out("ribo_only_v3.png"))
```



## RNA & Ribo synchronous

```
tbl_synchro <-
  tbl_input %>%
  dplyr::filter(te_morf_padj > 0.05,
    morf_BF_BH < 0.05, (morf_Max_Amp/morf_Max) > 0.5,
    rna_BF_BH < 0.05, (rna_Max_Amp/rna_Max) > 0.5) %>%
  dplyr::mutate(is_same_peak = rna_Phase == morf_Phase) %>%
  dplyr::select(AGI, rna_Phase, morf_Phase, is_same_peak,
    dplyr::matches("rna_coef_z"), dplyr::matches("^morf_coef_z"))
tbl_synchro %>% dplyr::glimpse()
```

```
Rows: 1,510
Columns: 14
$ AGI      <chr> "AT1G01210", "AT1G01370", "AT1G01420", "AT1G01500", "AT...
$ rna_Phase <dbl> 0, 6, 3, 6, 0, 6, 21, 0, 0, 12, 21, 9, 18, 3, 3, 0, 6, ...
$ morf_Phase <dbl> 21, 3, 6, 6, 0, 6, 18, 18, 0, 12, 18, 6, 18, 3, 3, 3, 3...
$ is_same_peak <lgl> FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, FALSE, FALSE, TR...
$ rna_coef_zt3 <dbl> -1.16790700, 0.44953541, 0.50432025, 1.88809027, -0.329...
```

```

$ rna_coef_zt6 <dbl> -1.05731323, 0.46647523, 0.46836302, 2.40909529, -0.555...
$ rna_coef_zt12 <dbl> -1.25621376, 0.14569048, -0.80404491, 1.64694275, -2.45...
$ rna_coef_zt18 <dbl> -0.37037688, -0.26925912, -0.80975832, 0.98976935, -3.7...
$ rna_coef_zt21 <dbl> -0.1543707, -0.5761637, -0.7721994, -0.3356448, -2.0136...
$ morf_coef_zt3 <dbl> -1.6449053, 0.5706022, 0.6611718, 2.0186820, -0.2468353...
$ morf_coef_zt6 <dbl> -1.41630178, 0.22085276, 0.98983105, 2.41271352, -0.267...
$ morf_coef_zt12 <dbl> -1.1781683, -0.2142790, 0.3271142, 2.1341795, -1.662285...
$ morf_coef_zt18 <dbl> 0.20814813, -0.70554507, -0.58096405, 1.00043471, -1.43...
$ morf_coef_zt21 <dbl> 0.27668458, -0.28433127, -0.40665455, -0.04852668, -0.7...

```

```

tbl_plot <-
  convert_to_tbl_plot(tbl_synchro, AGI) %>%
  scale_tbl_plot(scaling_func = centering_then_z_score) %>%
  sort_tbl(rna_Phase, peak_rna, morf_Phase, peak_morf)
gp_hm <- plot_heatmap(tbl_plot)

```

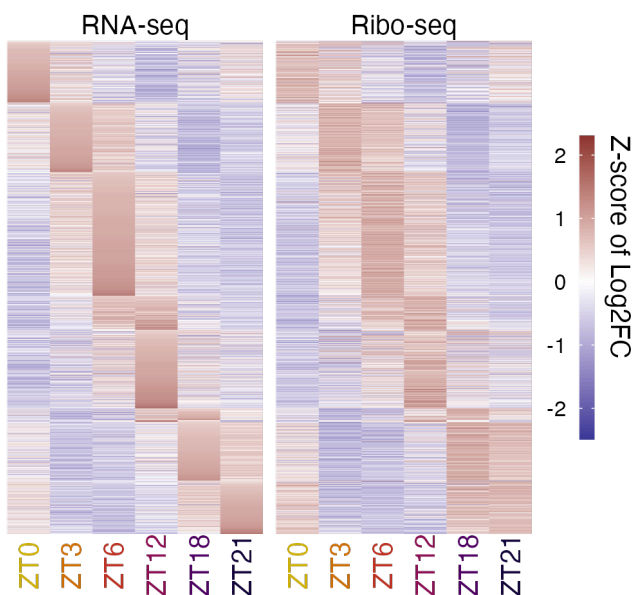
Warning: Vectorized input to `element\_text()` is not officially supported.  
 i Results may be unexpected or may change in future versions of ggplot2.

```

ggsave_single(gp_hm, filename = path_out("synchro_v3.png"), height = 80)
ggsave_single(gp_hm, filename = path_out("synchro_v3.svg"), height = 80)
ggsave_single(gp_hm + theme_empty(), filename = path_out("synchro_trim_v3.png"),
  width = 69, height = 65)

```

```
knitr::include_graphics(path_out("synchro_v3.png"))
```



## RNA & Ribo asynchronous (separate amplitude/phase)

```

tbl_asynchro <-
  tbl_input %>%
  dplyr::filter(te_morf_padj < 0.05,
    morf_BF_BH < 0.05, (morf_Max_Amp/morf_Max) > 0.5,
    rna_BF_BH < 0.05, (rna_Max_Amp/rna_Max) > 0.5) %>%
  dplyr::mutate(is_same_peak = rna_Phase == morf_Phase) %>%
  dplyr::select(AGI, rna_Phase, morf_Phase, is_same_peak,
    dplyr::matches("rna_coef_z"), dplyr::matches("^morf_coef_z"))
tbl_asynchro %>% dplyr::glimpse()

```



```

Rows: 2,486
Columns: 14
$ AGI          <chr> "AT1G01050", "AT1G01060", "AT1G01070", "AT1G01250", "AT...
$ rna_Phase    <dbl> 21, 0, 0, 6, 6, 18, 0, 18, 0, 21, 6, 21, 6, 21, 3, 12, ...
$ morf_Phase   <dbl> 18, 0, 0, 3, 6, 0, 3, 18, 0, 21, 6, 18, 6, 18, 6, 3, 0,...
$ is_same_peak <lgl> FALSE, TRUE, TRUE, FALSE, TRUE, FALSE, FALSE, TRUE, TRU...
$ rna_coef_zt3 <dbl> -0.6300576, -3.7463561, -0.2584560, 0.6506670, 1.092370...
$ rna_coef_zt6 <dbl> -0.69642283, -7.65539591, -1.02727619, 0.77227635, 2.12...
$ rna_coef_zt12 <dbl> -0.43469075, -12.22596633, -1.97221210, -0.42975742, 1...
$ rna_coef_zt18 <dbl> 0.250006103, -4.762324389, -0.828677015, -1.201335892, ...
$ rna_coef_zt21 <dbl> 0.55383625, -1.03071034, -0.11111277, -0.81296218, 0.13...
$ morf_coef_zt3 <dbl> -0.92368964, -4.31293835, -1.58757170, 2.18991004, 1.84...
$ morf_coef_zt6 <dbl> -0.9931011, -7.2137217, -2.5218907, 1.6253208, 2.150141...
$ morf_coef_zt12 <dbl> -0.8979904, -11.6701898, -2.4024669, 1.1480482, 1.91007...
$ morf_coef_zt18 <dbl> 0.59371601, -4.65761586, -0.79186508, -0.96892338, -0.2...
$ morf_coef_zt21 <dbl> 0.32666295, -0.71976503, -0.57452122, -0.43784508, -0.3...

```

```

tbl_plot <-
  convert_to_tbl_plot(tbl_asynchro, AGI) %>%
  scale_tbl_plot(scaling_func = centering_then_z_score) %>%
  sort_tbl(rna_Phase, peak_rna, morf_Phase, peak_morf)
gp_hm <-
  tbl_plot %>%
  dplyr::mutate(AGI = forcats::fct_inorder(AGI)) %>%
  dplyr::mutate(
    is_same_peak =
      ifelse(is_same_peak, "Amplitude", "Phase")
  ) %>%
  ggplot(aes(time, forcats::fct_rev(AGI))) +
  geom_tile(aes(fill = scaled)) +
  scale_y_discrete(expand = expansion(0)) +
  scale_fill_gradient2(low = scales::muted("blue"), high = scales::muted("red")) +
  facet_grid(rows = vars(is_same_peak), cols = vars(type),
    scales = "free_y", space = "free_y", switch = "y") +
  theme_void(base_size = 10) +
  theme(
    axis.text.x = element_text(angle = 90, color = LABEL_PALETTE),
    strip.text.x.top = element_text(size = 10, vjust = 1.1),
    strip.text.y.left = element_text(size = 10, angle = 90),
    legend.title = element_text(angle = -90),
    plot.margin = margin(10/2, 10/2, 10/2, 10/2),
    panel.spacing.x = unit(0, units = "mm"),
    panel.spacing.y = unit(1, units = "mm")
  ) +
  guides(fill = guide_colorbar(
    title = "Z-score of Log2FC",
    title.position = "right",
    barwidth = 2, barheight = 40, default.unit = "mm",
    label.position = "left", label.hjust = 1
  ))

```

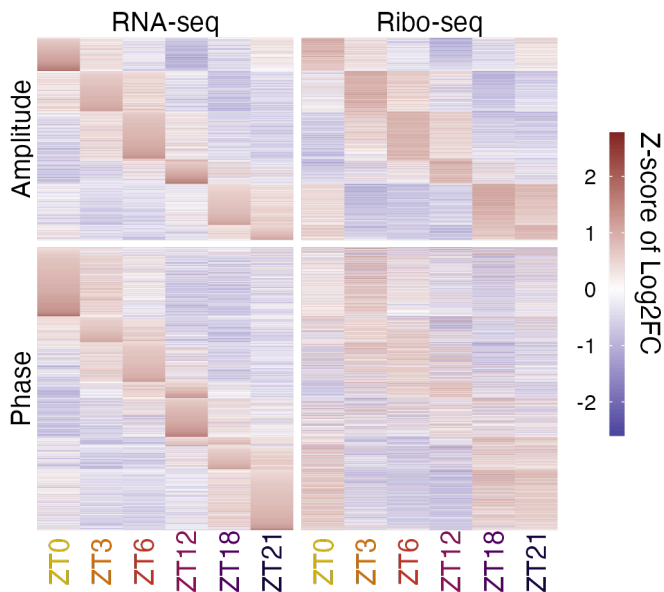
Warning: Vectorized input to `element\_text()` is not officially supported.  
 i Results may be unexpected or may change in future versions of ggplot2.

```

ggsave_single(filename = path_out("asynchro_v3.png"), gp_hm, width = 90, height = 80)
ggsave_single(filename = path_out("asynchro_v3.svg"), gp_hm, width = 90, height = 80)
ggsave_single(gp_hm + theme_empty(), filename = path_out("asynchro_trim_v3.png"),
  width = 69, height = 65)

```

```
knitr::include_graphics(path_out("asynchro_v3.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] tidyselect_1.2.0	xfun_0.40	purrr_1.0.1
[4] colorspace_2.0-3	vctrs_0.6.1	generics_0.1.3
[7] htmltools_0.5.3	viridisLite_0.4.1	yaml_2.3.6
[10] utf8_1.2.2	rlang_1.1.0	pillar_1.9.0
[13] glue_1.6.2	withr_2.5.0	bit64_4.0.5
[16] lifecycle_1.0.3	stringr_1.5.0	munsell_0.5.0
[19] gtable_0.3.1	ragg_1.2.5	evaluate_0.20
[22] labeling_0.4.2	knitr_1.42	forcats_1.0.0
[25] tzdb_0.3.0	fastmap_1.1.0	parallel_4.2.1
[28] fansi_1.0.3	readr_2.1.4	renv_1.0.3
[31] scales_1.2.1	BiocManager_1.30.18	vroom_1.6.0
[34] jsonlite_1.8.4	farver_2.1.1	fs_1.5.2
[37] systemfonts_1.0.4	bit_4.0.5	textshaping_0.3.6
[40] gridExtra_2.3	png_0.1-7	hms_1.1.3
[43] digest_0.6.31	stringi_1.7.12	dplyr_1.1.1
[46] grid_4.2.1	rprojroot_2.0.3	here_1.0.1
[49] cli_3.6.0	tools_4.2.1	tibble_3.2.1

[52]	crayon_1.5.2	tidyr_1.3.0	pkgconfig_2.0.3
[55]	svglite_2.1.0	rmarkdown_2.24	rstudioapi_0.14
[58]	viridis_0.6.2	R6_2.5.1	prismatic_1.1.1
[61]	compiler_4.2.1		