Figure 4, Figure 5, and Figure 6

Toshihiro Arae

General directory setting

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

Loading packages

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

Load common R scripts

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

Load script

```r
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
```

```r
#' @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

```
path_out_fig4 <- function(...) fs::path(wd, "analysis", "fig", "fig04", ...)
path_out_fig5 <- function(...) fs::path(wd, "analysis", "fig", "fig05", ...)
path_out_fig6 <- function(...) fs::path(wd, "analysis", "fig", "fig06", ...)
fs::dir_create(path_out_fig4())
fs::dir_create(path_out_fig5())
fs::dir_create(path_out_fig6())
```

Define some functions

Functions to create tibbles

```
# function to change pattern scaling
# choose either one
rel_to_ave <- function(x) x / mean(x)
rel_to_max <- function(x) x / max(x)

#' Extract normalized count of gene of interest and return a tibble for plot
#' @param GOI a character vector. AGI code of gene of interest
#'
make_tbl_plot <- function(GOI) {
  tbl_nc %>%
    dplyr::filter(AGI == GOI) %>%
    tidyr::pivot_longer(!AGI) %>%
    dplyr::mutate(
      ZT =
        stringr::str_extract(name, "zt\\d+") %>%
        stringr::str_to_upper() %>%
        forcats::fct_relevel(names(COL_PALETTE)),
      type =
        stringr::str_extract(name, "rna|morf|uorf") %>%
        {c(rna = "RNA", morf = "mORF", uorf = "uORF")[.]} %>%
        forcats::fct_relevel(c("RNA", "mORF", "uORF"))
    ) %>%
    dplyr::with_groups(c(type), dplyr::mutate, rel = rel_to_ave(value))
}

#' Extract effect size of gene of interest and return a tibble for plot
#' @param GOI a character vector. AGI code of gene of interest
#'
make_tbl_plot_coef <- function(tbl, GOI) {
  if(nrow(tbl) == 0) {
    return(
      tibble::tibble(
        AGI = GOI,
        ZT = paste0("ZT", c(0,3,6,12,18,21)) %>%
          forcats::fct_relevel(names(COL_PALETTE)),
        coef = NA, se = NA
      )
    )
  }
  tbl %>%
    dplyr::select(AGI, coef_zt3:coef_zt21, se_zt3:se_zt21) %>%
    dplyr::mutate(coef_zt0 = 0, .before = coef_zt3) %>%
    dplyr::mutate(se_zt0 = 0, .before = se_zt3) %>%
    tidyr::pivot_longer(!AGI) %>%
    dplyr::mutate(
      ZT =
        stringr::str_extract(name, "zt\\d+") %>%
        stringr::str_to_upper() %>%
        forcats::fct_relevel(names(COL_PALETTE)),
      type = stringr::str_extract(name, "coef|se")
```

```r
    ) %>%
    dplyr::select(!name) %>%
    tidyr::pivot_wider(names_from = type, values_from = value)
}
```

Global constants

```r
SIZE_POINT <- 2.5

WIDTH_DODGE <- 2/3

PAL_LINECOLOR <-
  c(RNA = "red", mORF = "blue", uORF = "green") %>%
  scales::muted(l = 50, c = 80)
```

Functions to plot

```r
#' Function to label numbers in the specific format
label_num <- function(x) {
  # signs::signs(x, accuracy = .1, add_plusses = TRUE, label_at_zero = "symbol") %>%
  signs::signs(x, accuracy = .01, add_plusses = TRUE, label_at_zero = "symbol") %>%
    chartr("+±−", "  -", .)
}

#' Partialized `ggsave()`
ggsave_ <- purrr::partial(ggsave, width = 80, height = 80, dpi = 300, units = "mm")

#' Add some plot layers of normalized count
#' @param tbl
#' @param x
#'
plot_nc <- function(tbl, x) {
  force(tbl)

  list(
    geom_line(
      data =
        tbl %>%
        dplyr::group_by(type, {{x}}, facet) %>%
        dplyr::summarise(ave = mean(rel), .groups = "drop") %>%
        dplyr::ungroup(),
      mapping = aes(x = {{x}}, y = ave, color = type)
    ),
    geom_point(
      data = tbl,
      mapping = aes(
        x = {{x}}, y = rel,
        color = stage(type, after_scale = alpha(color, .5)),
        shape = type),
      size = SIZE_POINT
    ),
    scale_color_manual(values = PAL_LINECOLOR)
  )
}

#' Add some plot layers of estimated coefficients (effect size)
#' @param tbl
#' @param x
#'
plot_te <- function(tbl, x, scale_to_x, scale_to_y) {
  force(tbl)
  force(scale_to_x)
  force(scale_to_y)

  list(
    geom_line(
      data = tbl,
      aes(x = {{x}}, y = scale_to_x(coef),
          group = type, linetype = type),
      position = position_dodge(width = WIDTH_DODGE)
    ),
    geom_pointrange(
      data = tbl,
      aes(x = {{x}}, y = scale_to_x(coef),
          ymin = scale_to_x(coef - se),
          ymax = scale_to_x(coef + se),
          group = type),
      shape = 15,
      position = position_dodge(width = WIDTH_DODGE)
    ),
    scale_y_continuous(
      expand = expansion(mult = c(0, .1)),
      labels = label_num,
      sec.axis = sec_axis(
        name = "Log<sub>2</sub>(FC) (vs. TE at ZT0)",
        trans = scale_to_y,
        labels = label_num
      )
    )
  )
}
```

Load input data

Normalized count and log2FC

```
# Load normalized counts -------------------------------------
dir_readcount <- fs::path("analysis", "list_summary")

tbl_nc <-
  fs::dir_ls(fs::path(wd, dir_readcount), regexp = "_nc.csv") %>%
  purrr::map(readr::read_csv, show_col_types = FALSE) %>%
  purrr::reduce(dplyr::left_join, by = "AGI")

# Load DESeq2 results -------------------------------------
tbl_out <-
  readr::read_csv(fs::path(wd, "analysis", "deseq2_te", "deg_all.csv"),
                  show_col_types = FALSE)
tbl_out_uorf <-
  readr::read_csv(fs::path(wd, "analysis", "deseq2_te_uorf", "deg_all.csv"),
                  show_col_types = FALSE)
```

Plotting

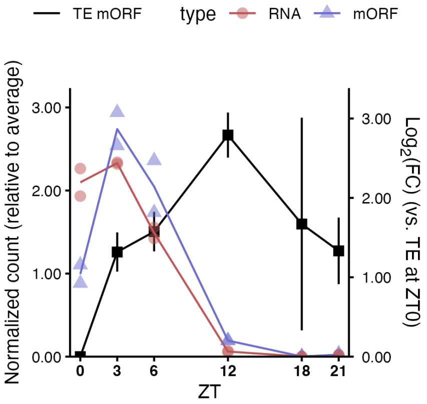
```
theme_fig04 <- function(base_size = 10, base_line_size = 1/22) {
  list(
    theme_linedraw(
      base_size = base_size,
      base_line_size = base_line_size
    ),
    labs(x = "ZT", y = "Normalized count (relative to average)"),
    theme(
      axis.text.x = element_text(face = "bold"),
      axis.title.y.left = ggtext::element_markdown(),
      axis.title.y.right = ggtext::element_markdown(),
      axis.text.y.left = element_text(hjust = 1),
      axis.text.y.right = element_text(hjust = 1),
      panel.border = element_blank(),
      axis.line = element_line(color = "black", linewidth = .5),
      panel.grid = element_blank(),
      axis.ticks = element_line(linewidth = .5),
      legend.position = "top"
    ),
    guides(
      alpha = guide_legend("", order = 1),
      linetype = guide_legend("", order = 2),
      fill = guide_none()
    )
  )
}
```

Figure 4

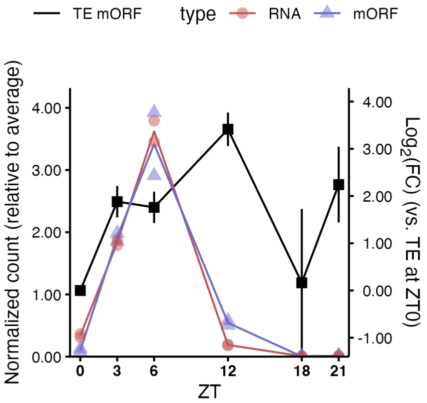
```
tbl_goi <-
  tibble::tribble(
    ~AGI,        ~gene_name,
    "AT5G64170", "LNK1",
    "AT5G02810", "PRR7",
    "AT1G22770", "GI",
    "AT2G46790", "PRR9",
    "AT3G46640", "LUX",
    "AT5G24470", "PRR5",
    "AT5G60100", "PRR3",
    "AT2G40080", "ELF4",
    "AT5G57360", "ZTL",
  )

plot_over_tbl_goi <- function(tbl_goi, path_out) {
  li_gp1 <- list()
  for(i in seq_along(tbl_goi$AGI)) {
    temp_GOI <- tbl_goi$AGI[i]
    temp_name <- tbl_goi$gene_name[i] %>% stringr::str_replace_all("/", "_")
    temp_name <- ifelse(is.na(temp_name), "", paste0("_", temp_name))
    label <- stringr::str_glue("{temp_GOI}{temp_name}")

    tbl_plot <-
      make_tbl_plot(temp_GOI) %>%
      dplyr::filter(type != "uORF")

    tbl_plot_coef <-
      dplyr::bind_rows(
        tbl_out %>%
          dplyr::filter(AGI == temp_GOI) %>%
          make_tbl_plot_coef(temp_GOI) %>%
          dplyr::mutate(type = "TE mORF")
      )

    if(all(is.na(tbl_plot$value) | tbl_plot$value == 0))
      next

    y1 <- tbl_plot$rel
    y2 <- with(tbl_plot_coef, {c(coef+se, coef-se)})
```

```r
    scale_to_y <- make_scale_y1_to_y2(y1, y2)
    scale_to_x <- make_scale_y2_to_y1(y1, y2)
    scale_to_x_se <- make_scale_y2_to_y1_se(y1, y2)

    gp1 <-
      ggplot() +
      tbl_plot_coef %>%
      dplyr::mutate(zt = as.integer(stringr::str_extract(ZT, "\\d+"))) %>%
      plot_te(x = zt, scale_to_x, scale_to_y) +
      tbl_plot %>%
      dplyr::mutate(zt = as.integer(stringr::str_extract(ZT, "\\d+"))) %>%
      dplyr::mutate(facet = "count") %>%
      plot_nc(zt) +
      scale_x_continuous("ZT", breaks = c(0, 3, 6, 12, 18, 21),
                         expand = expansion(add = .5)) +
      facet_grid(cols = vars(facet), scales = "free_y") +
      theme_fig04() +
      theme(strip.background = element_blank())
    li_gp1[[label]] <- gp1

    force(scale_to_x(1:10))
    force(scale_to_y(1:10))
  }

  max_dims1 <- patchwork::get_max_dim(li_gp1)

  for(i in seq_along(tbl_goi$AGI)) {
    temp_GOI <- tbl_goi$AGI[i]
    temp_name <- tbl_goi$gene_name[i]
    label <- stringr::str_glue("{sprintf('%02d', i)}_{temp_GOI}_{temp_name}")

    gp1 <- patchwork::set_dim(li_gp1[[i]], max_dims1)
    ggsave_(path_out(stringr::str_glue("{label}.png")), gp1)
    # ggsave_(path_out(stringr::str_glue("{label}.svg")), gp1)
    # ggsave_(path_out(stringr::str_glue("{label}.pdf")), gp1)
  }
}

plot_over_tbl_goi(tbl_goi = tbl_goi, path_out = path_out_fig4)
```

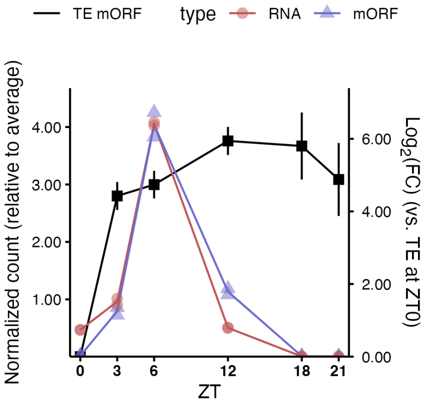
01_AT5G64170_LNK1.png



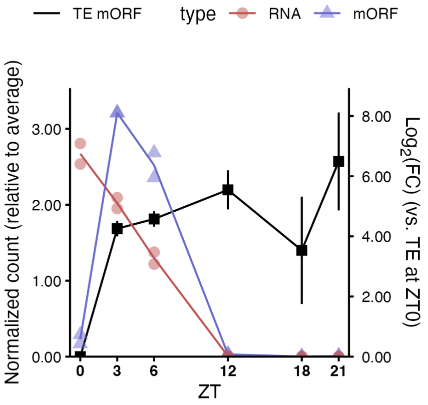
02_AT5G02810_PRR7.png



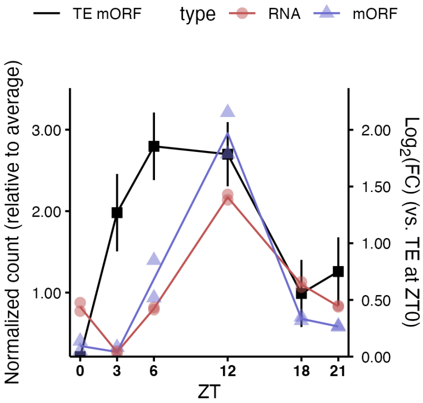
03_AT1G22770_GI.png



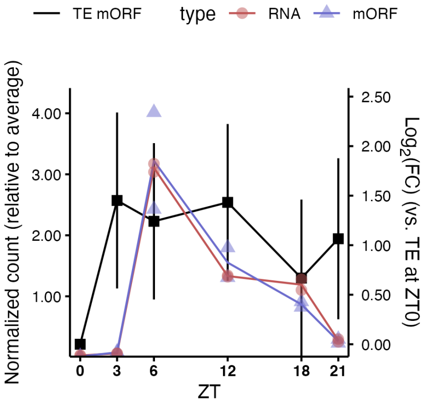
04_AT2G46790_PRR9.png



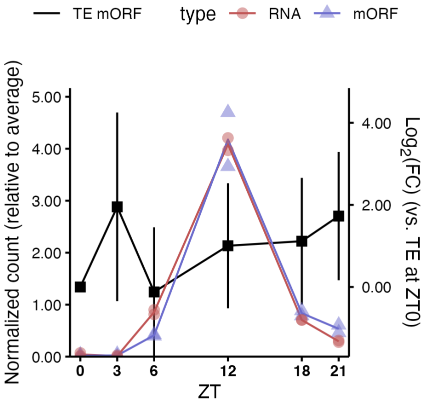
05_AT3G46640_LUX.png



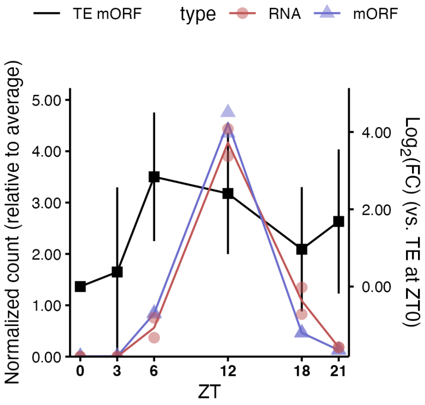
06_AT5G24470_PRR5.png



07_AT5G60100_PRR3.png



08_AT2G40080_ELF4.png



09_AT5G57360_ZTL.png

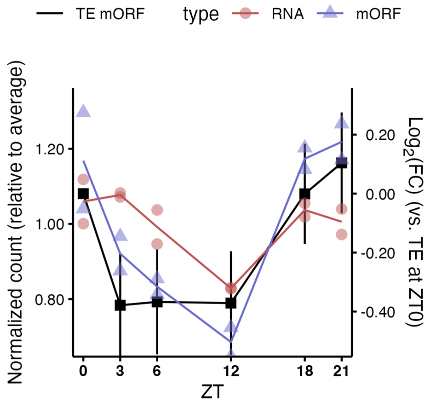
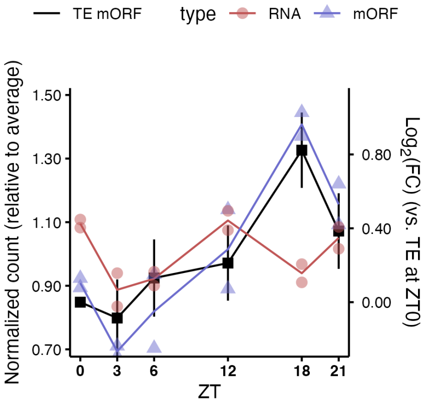


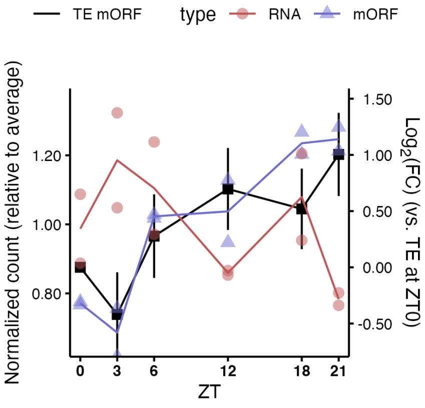
Figure 5

```
tbl_goi <-
  tibble::tribble(
    ~AGI,        ~gene_name,
    "AT1G07670", "ECA4",
    "AT1G08960", "CAX11",
    "AT1G10130", "ECA3",
    "AT1G53210", "NCL",
    "AT2G17780", "MCA2",
    "AT2G23790", "MCU3",
    "AT2G32400", "GLR3.7",
    "AT3G13320", "CAX2",
    "AT3G63380", "ACA12",
    "AT4G00900", "ECA2",
  )

plot_over_tbl_goi(tbl_goi = tbl_goi, path_out = path_out_fig5)
```

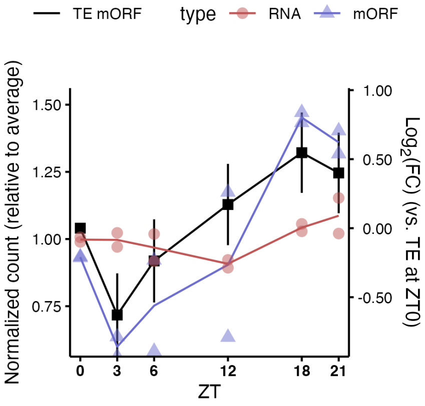
01_AT1G07670_ECA4.png



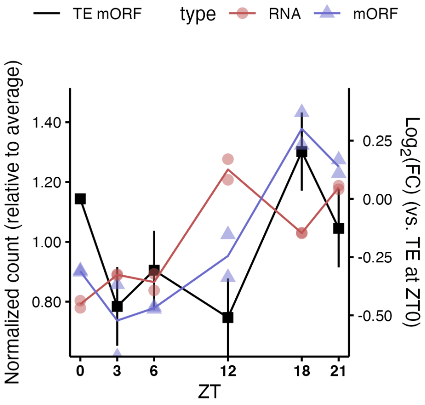
02_AT1G08960_CAX11.png



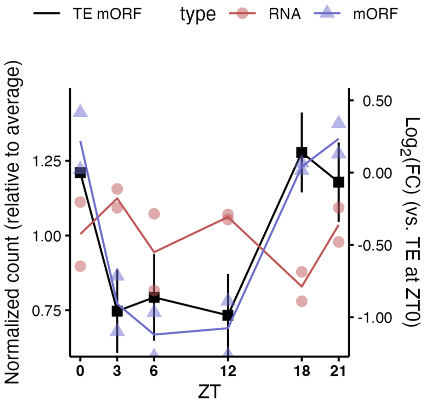
03_AT1G10130_ECA3.png



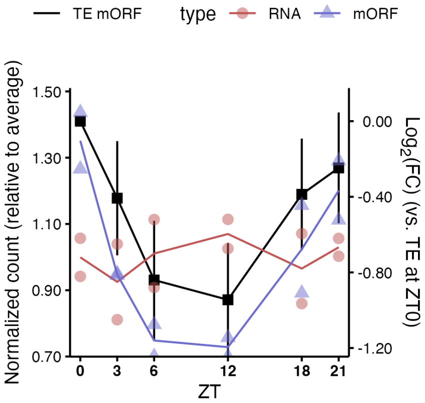
04_AT1G53210_NCL.png



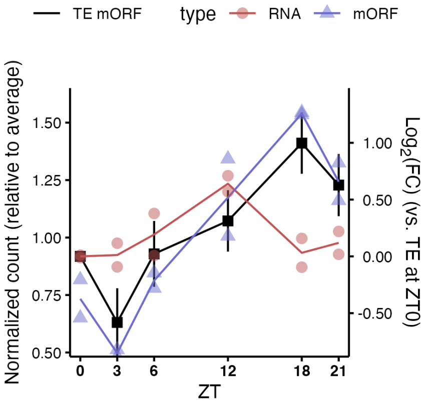
05_AT2G17780_MCA2.png



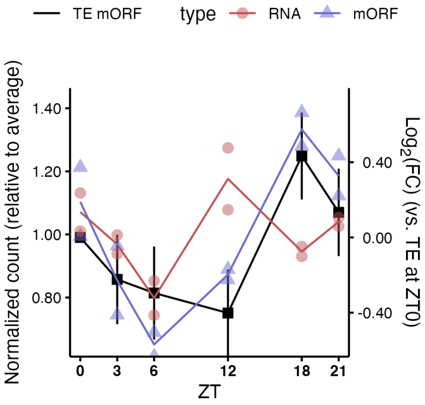
06_AT2G23790_MCU3.png



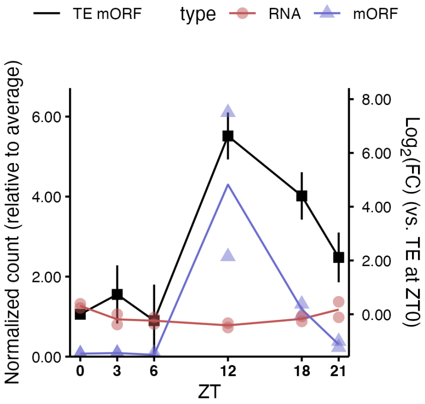
07_AT2G32400_GLR3.7.png



08_AT3G13320_CAX2.png



09_AT3G63380_ACA12.png



10_AT4G00900_ECA2.png

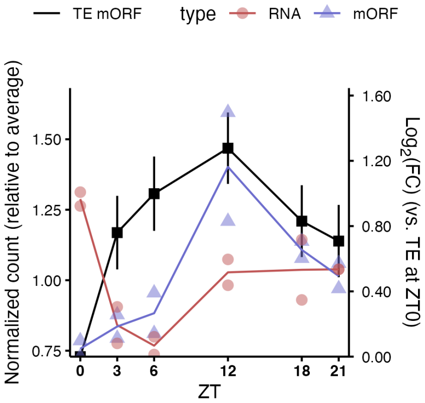
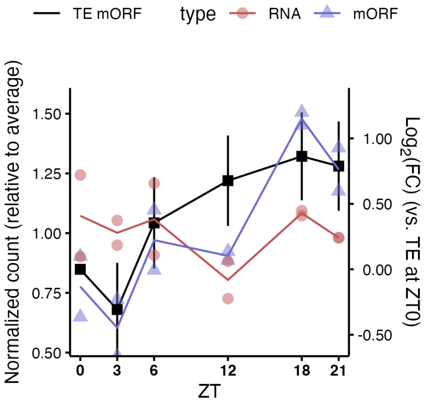


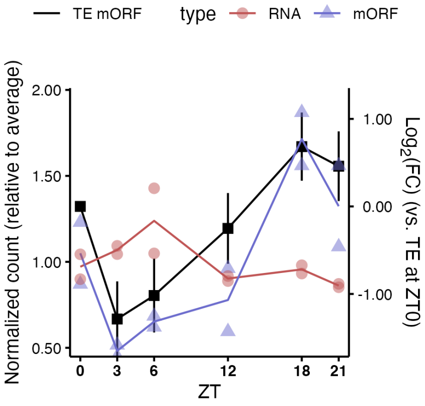
Figure 6

```r
tbl_goi <-
  tibble::tribble(
    ~AGI,         ~gene_name,
    "AT5G38460", "ALG6",
    "AT5G02410", "ALG10",
    "AT4G21150", "HAP6",
    "AT1G51590", "MNS1",
    "AT1G30000", "MNS3",
    "AT5G43710", "MNS4",
    "AT1G74340", "DPMS2",
    "AT1G76400", "OST1B",
    "AT1G16570", "TUN",
  )

plot_over_tbl_goi(tbl_goi = tbl_goi, path_out = path_out_fig6)
```

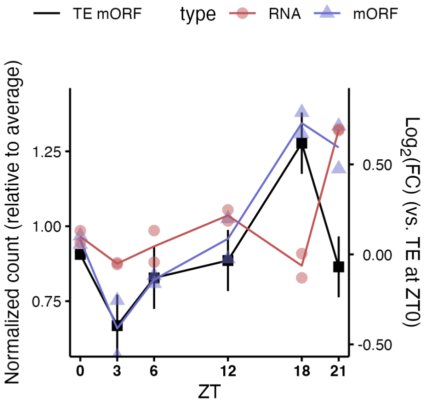
01_AT5G38460_ALG6.png



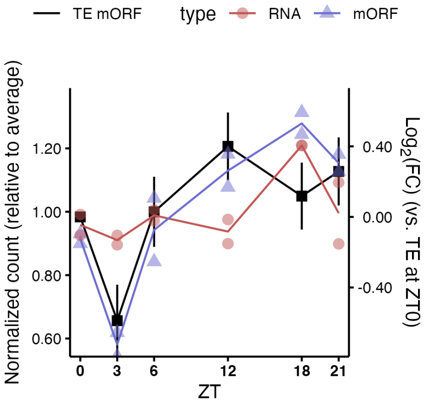
02_AT5G02410_ALG10.png



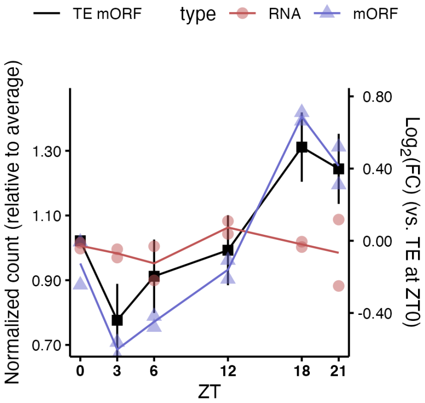
03_AT4G21150_HAP6.png



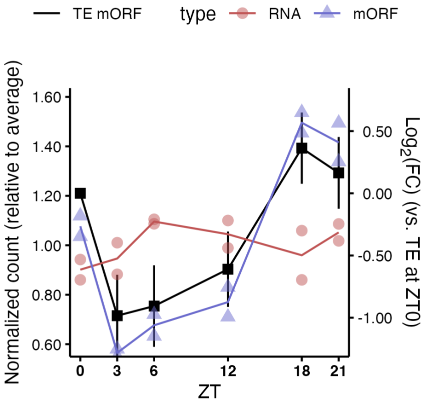
04_AT1G51590_MNS1.png



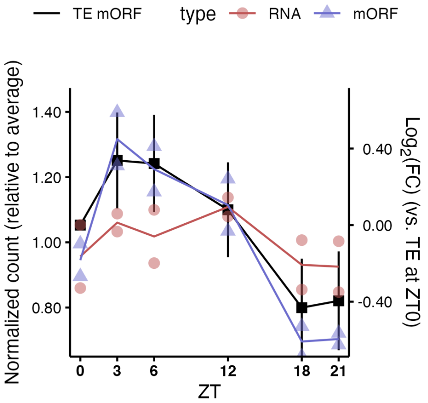
05_AT1G30000_MNS3.png



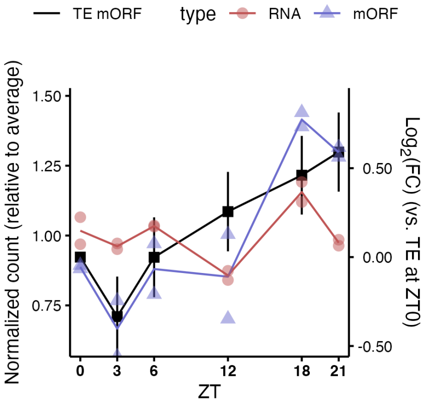
06_AT5G43710_MNS4.png



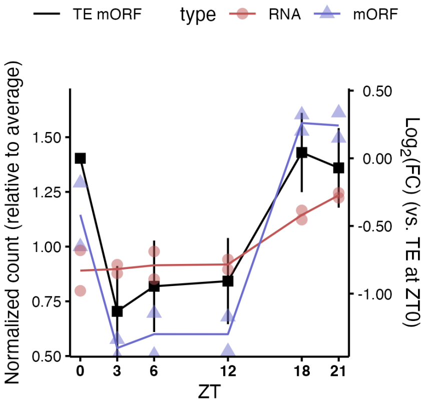
07_AT1G74340_DPMS2.png



08_AT1G76400_OST1B.png



09_AT1G16570_TUN.png



Sessioninfo

```r
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         gridtext_0.1.5
[13] pillar_1.9.0        glue_1.6.2          withr_2.5.0
[16] bit64_4.0.5         lifecycle_1.0.3     stringr_1.5.0
[19] commonmark_1.9.0    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      markdown_1.9
[31] fansi_1.0.3         signs_0.1.2         Rcpp_1.0.11
[34] readr_2.1.4         renv_1.0.3          scales_1.2.1
[37] BiocManager_1.30.18 magick_2.7.3        vroom_1.6.0
[40] jsonlite_1.8.4      systemfonts_1.0.4   farver_2.1.1
[43] fs_1.5.2            bit_4.0.5           textshaping_0.3.6
[46] gridExtra_2.3       hms_1.1.3           digest_0.6.31
[49] stringi_1.7.12      dplyr_1.1.1         grid_4.2.1
[52] rprojroot_2.0.3     here_1.0.1          cli_3.6.0
[55] tools_4.2.1         patchwork_1.1.2     tibble_3.2.1
[58] crayon_1.5.2        tidyr_1.3.0         pkgconfig_2.0.3
[61] xml2_1.3.3          rmarkdown_2.24      rstudioapi_0.14
[64] viridis_0.6.2       R6_2.5.1            prismatic_1.1.1
[67] ggtext_0.1.2        compiler_4.2.1
```