Figure 3B

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"))
```

Directory setting

```
dir_output <- fs::path("analysis", "fig", "fig03")
path_out <- function(...) fs::path(wd, dir_output, ...)
fs::dir_create(path_out())
```

Prepare input data

```
inf <- fs::path(wd, "analysis", "list_summary", "summary_all.csv")
tbl_input <-
  readr::read_csv(inf) %>%
  dplyr::filter(!grepl("AT[CM]G", AGI)) %>%
  dplyr::filter(!is.na(te_morf_padj)) %>%
  dplyr::arrange(rna_Phase, morf_Phase)
```

```
Rows: 33341 Columns: 330
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (10): AGI, locus_type, symbol, alias, full_name, curator_summary, descr...
dbl (295): zt0_1_rna, zt0_2_rna, zt12_1_rna, zt12_2_rna, zt18_1_rna, zt18_2_...
lgl  (25): rna_allZero, rna_dispOutlier, rna_fullBetaConv, rna_reducedBetaCo...

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.
```

```
temp_bg <- tbl_input$AGI

li_tbl_input <-
  list(
    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),
    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),
    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),
    asynchro_phase =
      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::filter(rna_Phase != morf_Phase),
    asynchro_amplitude =
      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::filter(rna_Phase == morf_Phase)
  )

li_tbl_input$rnaonly <-
  li_tbl_input$rnaonly %>%
  split(.$rna_Phase) %>%
  {c(list("all" = li_tbl_input$rnaonly), .)}

li_tbl_input$riboonly <-
  li_tbl_input$riboonly %>%
  split(.$morf_Phase) %>%
  {c(list("all" = li_tbl_input$riboonly), .)}

li_tbl_input$synchro <-
  li_tbl_input$synchro %>%
  split(.$rna_Phase) %>%
  {c(list("all" = li_tbl_input$synchro), .)}

li_tbl_input$asynchro_amplitude <-
  li_tbl_input$asynchro_amplitude %>%
  # split(.$rna_Phase)
  split(.$morf_Phase) %>%
  {c(list("all" = li_tbl_input$asynchro_amplitude), .)}

li_tbl_input$asynchro_phase <-
  li_tbl_input$asynchro_phase %>%
  split(.$morf_Phase) %>%
  {c(list("all" = li_tbl_input$asynchro_phase), .)}

li_tbl_input <-
  li_tbl_input %>%
  unlist(recursive = FALSE) %>%
  purrr::keep(.p = ~ nrow(.x) >= 10)
```

Perform GOterm enrichment test

```
for(i in seq_along(li_tbl_input)) {
  temp_fg <- li_tbl_input[[i]] %>% dplyr::pull(AGI)
  temp_label <- names(li_tbl_input)[i]
  ngsmisc::clP_write_li_ego(
    fg = temp_fg,
    bg = temp_bg,
    OrgDb = "org.At.tair.db",
    keyType = "TAIR",
    pAdjustMethod = "BH",
    pvalueCutoff = 1,
    qvalueCutoff = 0.05,
    readable = FALSE,
    out_dir = path_out(""),
    label = temp_label
  )
}
```

Load GOterm enrichment test results

```
rds_files <-
  fs::path(path_out(), "ego_rds") %>%
  fs::dir_ls(regexp = ".rds$") %>%
  purrr::set_names(~ fs::path_file(.x))
```

Filter GOterm by semantic similalities

```
li_semsim <-
  list(
    BP = GOSemSim::godata("org.At.tair.db", ont = "BP"),
    CC = GOSemSim::godata("org.At.tair.db", ont = "CC"),
    MF = GOSemSim::godata("org.At.tair.db", ont = "MF")
  )
```

```
preparing gene to GO mapping data...
```

```
preparing IC data...
```

```
preparing gene to GO mapping data...
```

```
preparing IC data...
```

```
preparing gene to GO mapping data...
```

```
preparing IC data...
```

```
CUTOFF <- .7
goterm_clustering <- function(GOID, cutoff) {
  ONT <- clusterProfiler::go2ont(GOID[1])$Ontology
  if(length(GOID) > 1) {
    tbl <-
      simplifyEnrichment::GO_similarity(GOID, ont = ONT, measure = "Wang",
                                        db = "org.At.tair.db") %>%
      simplifyEnrichment::simplifyGO(
        method = "binary_cut",
        control = list(cutoff = cutoff),
        plot = FALSE
      ) %>%
      tibble::as_tibble()
  } else {
    tbl <-
      tibble::tibble(
        id = GOID,
        cluster = 1
      )
  }

  if(ONT == "BP") {
    ENV <- GO.db::GOBPANCESTOR
  } else if(ONT == "CC") {
    ENV <- GO.db::GOCCANCESTOR
  } else {
    ENV <- GO.db::GOMFANCESTOR
  }

  tbl %>%
    dplyr::mutate(
      ancestor = purrr::map(id, AnnotationDbi::get, env = ENV)
    ) %>%
    dplyr::rename(ID = id)
}

tbl_data <-
  rds_files %>%
  purrr::map(readRDS) %>%
  purrr::map("BP") %>%
  purrr::map(tibble::as_tibble) %>%
  purrr::keep(~ nrow(.x) > 0) %>%
  purrr::map(dplyr::filter, qvalue < 0.05) %>%
  purrr::keep(~ nrow(.x) > 0) %>%
  purrr::imap(~ dplyr::mutate(.x, sample = .y)) %>%
  dplyr::bind_rows() %>%
  dplyr::mutate(
    group =
      stringr::str_remove(sample, "li_ego_") %>%
      stringr::str_replace(".rds", "")
  )
```

```
Loading required package: DOSE
```

```
DOSE v3.22.1  For help: https://yulab-smu.top/biomedical-knowledge-mining-book/

If you use DOSE in published research, please cite:
Guangchuang Yu, Li-Gen Wang, Guang-Rong Yan, Qing-Yu He. DOSE: an R/Bioconductor package for Disease
Ontology Semantic and Enrichment analysis. Bioinformatics 2015, 31(4):608-609
```

```
tbl_data <-
  dplyr::left_join(
    tbl_data,
    goterm_clustering(tbl_data$ID, CUTOFF),
    by = "ID"
  )
```

```
```

```
Cluster 415 terms by 'binary_cut'... 60 clusters, used 0.401545 secs.
```

```
tbl_plot <-
  tbl_data %>%
  split(.$group) %>%
  purrr::map(dplyr::group_by, cluster) %>%
  purrr::map(dplyr::filter, qvalue == min(qvalue)) %>%
  purrr::map(~ dplyr::filter(.x, !(ID %in% unlist(ancestor)))) %>%
  purrr::map(dplyr::ungroup) %>%
  dplyr::bind_rows()
```

Setting font

```
extrafont::fonttable()
library(showtext)
sysfonts::font_add(
  family = "Arial Narrow",
  regular = "/System/Library/Fonts/Supplemental/Arial Narrow.ttf",
  bold = "/System/Library/Fonts/Supplemental/Arial Narrow Bold.ttf",
  italic = "/System/Library/Fonts/Supplemental/Arial Narrow Italic.ttf",
  bolditalic = "/System/Library/Fonts/Supplemental/Arial Narrow Bold Italic.ttf"
)
showtext::showtext_auto()
showtext::showtext_auto(enable = FALSE)
```

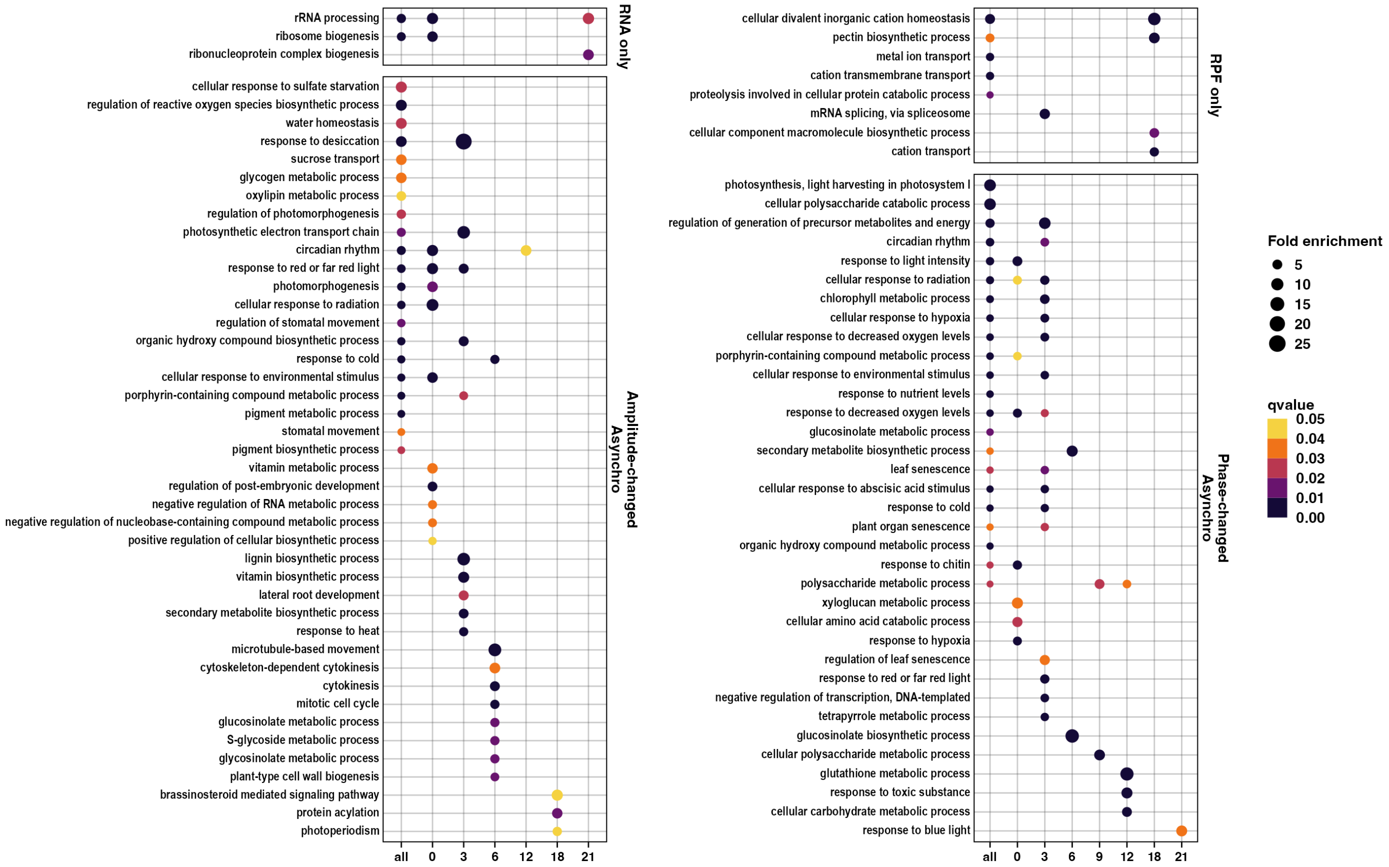
```
library(extrafont)
extrafont::loadfonts()
```

Setting plot theme

```
AXIS_TEXT_SIZE <- 6
theme_fig03 <- function(base_size = 10, base_line_size = 10/22) {
  list(
    theme_linedraw(
      base_size = base_size,
      base_line_size = base_line_size
    ),
    scale_fill_viridis_b(option = "B", limits = c(0, .05), begin = .1, end = .9),
    scale_color_viridis_b(option = "B", limits = c(0, .05), begin = .1, end = .9),
    scale_size(name = "Fold enrichment",
               breaks = scales::pretty_breaks(n = 5), range = c(.5, 3),
               limits = c(1, 25)),
    labs(x = "", y = ""),
    theme(
      plot.background = element_blank(),
      plot.margin = margin(),
      strip.text.y.right = element_text(color = "black", face = "bold", size = 7,
                                        margin = margin(r = .5, l = .5, unit = "mm")),
      strip.background = element_blank(),
      strip.clip = "off",
      axis.title.x = element_text(face = "bold", size = 7),
      axis.text.x = element_text(face = "bold", size = AXIS_TEXT_SIZE),
      axis.text.y = ggtext::element_markdown(family = "Arial Narrow",
                                             face = "bold",
                                             size = AXIS_TEXT_SIZE,
                                             margin = unit(c(0,-1,0,0), "pt")),
      axis.ticks.x = element_line(linewidth = .3, color = "black"),
      axis.ticks.y = element_blank(),
      legend.key.size = unit(c(3.3, 3.3), "mm"),
      legend.box.margin = margin(),
      legend.text = element_text(face = "bold", size = 7),
      legend.title = element_text(face = "bold", size = 7),
      panel.grid = element_blank(),
      panel.grid.major =
        element_line(linewidth = .3, color = alpha("black", .2))
    )
  )
}
```

Plot dot plot

```
WRAP_WIDTH <- 100
TH_COUNT <- 5

calc_mm <- function(x) sum(grid::convertUnit(x, "mm"))

lev_category <-
  c(
    "rnaonly" = "RNA only",
    "asynchro_amplitude" = "Amplitude-changed\nAsynchro",
    "riboonly" = "RPF only",
    "asynchro_phase" = "Phase-changed\nAsynchro"
    )

tbl_plot <-
  tbl_plot %>%
  dplyr::arrange(qvalue) %>%
  dplyr::mutate(
    term =
      Description %>%
      stringr::str_wrap(width = WRAP_WIDTH) %>%
      stringr::str_replace("\n", "<br>") %>%
      forcats::fct_inorder() %>%
      forcats::fct_rev(),
    a = stringr::str_extract(GeneRatio, "(\\d+)/(\\d+)", 1) %>% as.integer(),
    b = stringr::str_extract(GeneRatio, "(\\d+)/(\\d+)", 2) %>% as.integer(),
    c = stringr::str_extract(BgRatio, "(\\d+)/(\\d+)", 1) %>% as.integer(),
    d = stringr::str_extract(BgRatio, "(\\d+)/(\\d+)", 2) %>% as.integer(),
    fold_enrich = (a/b) / (c/d),
    category = stringr::str_extract(group, "^([^.]+)[.]", group = 1),
    zt =
      stringr::str_extract(group, "^[^.]+[.](.+)$", group = 1) %>%
      forcats::fct_relevel(c("all", as.character(seq(0, 21, 3)))),
  ) %>%
  dplyr::filter(Count >= TH_COUNT) %>%
  dplyr::arrange(zt)

tbl_plot <-
  tbl_plot %>%
  dplyr::with_groups(c(category, ID, cluster), tidyr::nest) %>%
  dplyr::filter(category != "synchro") %>%
  dplyr::mutate(
    category = forcats::fct_relevel(lev_category[category], lev_category)
  ) %>%
  dplyr::arrange(category, cluster) %>%
  dplyr::mutate(fcol = as.character(dplyr::row_number() %/% 46), .before = 1) %>%
  tidyr::unnest(cols = data) %>%
  dplyr::with_groups(c(category, ID, zt, fold_enrich, qvalue), tidyr::nest) %>%
  dplyr::with_groups(
    c(category, ID),
    dplyr::mutate,
    max_zt = forcats::fct_relevel(zt[which.max(fold_enrich)], "all"),
    has_all = any(zt == "all"),
    all_foldenrich = ifelse(has_all, fold_enrich[zt == "all"], NA)
  ) %>%
  split(paste0(.$has_all, .$category)) %>%
  {c(
    .[5:8] %>% purrr::map(~ dplyr::arrange(.x, category, desc(all_foldenrich), max_zt, qvalue)),
    .[1:4] %>% purrr::map(~ dplyr::arrange(.x, category, max_zt, desc(fold_enrich), qvalue))
  )} %>%
  dplyr::bind_rows() %>%
  tidyr::unnest(cols = data)

pgp <-
  patchwork::wrap_plots(
    ggplot(
      dplyr::filter(tbl_plot, fcol == 0) %>%
        dplyr::mutate(term = forcats::fct_inorder(term) %>% forcats::fct_rev()),
      aes(zt, term)) +
      geom_point(aes(color = qvalue, size = fold_enrich)),
    ggplot(
      dplyr::filter(tbl_plot, fcol == 1) %>%
        dplyr::mutate(term = forcats::fct_inorder(term) %>% forcats::fct_rev()),
      aes(zt, term)) +
      geom_point(aes(color = qvalue, size = fold_enrich)),
    ncol = 2, guides = "collect") &
  theme_fig03() &
  ggforce::facet_col(~ category, scales = "free_y", space = "free", strip.position = "right")
ggsave_ <- purrr::partial(ggsave, width = unit(240, "mm"),
                          height = unit(150, "mm"), units = "mm")
ggsave_(path_out("merged_sort_v2.png"), pgp)
ggsave_(path_out("merged_sort_v2.svg"), pgp)
ggsave_(path_out("merged_sort_v2.pdf"), pgp, device = cairo_pdf)
```



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] extrafont_0.19 DOSE_3.22.1    ggplot2_3.4.2  magrittr_2.0.3

loaded via a namespace (and not attached):
  [1] shadowtext_0.1.2         circlize_0.4.15          fastmatch_1.1-3
  [4] systemfonts_1.0.4        plyr_1.8.7               igraph_1.3.5
  [7] lazyeval_0.2.2           proxyC_0.3.3             splines_4.2.1
 [10] BiocParallel_1.30.4      GenomeInfoDb_1.32.4      digest_0.6.31
 [13] foreach_1.5.2            yulab.utils_0.0.5        htmltools_0.5.3
 [16] GOSemSim_2.22.0          viridis_0.6.2            GO.db_3.15.0
 [19] fansi_1.0.3              memoise_2.0.1            tm_0.7-11
 [22] cluster_2.1.3            doParallel_1.0.17        tzdb_0.3.0
 [25] ComplexHeatmap_2.12.1    Biostrings_2.64.1        readr_2.1.4
 [28] graphlayouts_1.0.1       RcppParallel_5.1.7       matrixStats_0.62.0
 [31] vroom_1.6.0              svglite_2.1.0            extrafontdb_1.0
 [34] enrichplot_1.16.2        colorspace_2.0-3         blob_1.2.3
 [37] ggrepel_0.9.4            textshaping_0.3.6        xfun_0.40
 [40] dplyr_1.1.1              crayon_1.5.2             RCurl_1.98-1.9
 [43] jsonlite_1.8.4           scatterpie_0.1.8         iterators_1.0.14
 [46] ape_5.6-2                glue_1.6.2               polyclip_1.10-4
 [49] gtable_0.3.1             zlibbioc_1.42.0          XVector_0.36.0
 [52] GetoptLong_1.0.5         Rttf2pt1_1.3.12          shape_1.4.6
 [55] BiocGenerics_0.42.0      scales_1.2.1             DBI_1.1.3
 [58] Rcpp_1.0.11              gridtext_0.1.5           viridisLite_0.4.1
 [61] clue_0.3-62              gridGraphics_0.5-1       tidytree_0.4.1
 [64] bit_4.0.5                stats4_4.2.1             httr_1.4.5
 [67] fgsea_1.22.0             RColorBrewer_1.1-3       pkgconfig_2.0.3
 [70] farver_2.1.1             utf8_1.2.2               here_1.0.1
 [73] labeling_0.4.2           ggplotify_0.1.0          tidyselect_1.2.0
 [76] rlang_1.1.0              reshape2_1.4.4           AnnotationDbi_1.58.0
 [79] munsell_0.5.0            tools_4.2.1              cachem_1.0.6
 [82] downloader_0.4           cli_3.6.0                generics_0.1.3
 [85] RSQLite_2.2.18           evaluate_0.20            stringr_1.5.0
 [88] fastmap_1.1.0            ragg_1.2.5               yaml_2.3.6
 [91] ggtree_3.4.4             org.Hs.eg.db_3.15.0      knitr_1.42
 [94] bit64_4.0.5              fs_1.5.2                 tidygraph_1.2.2
 [97] purrr_1.0.1              KEGGREST_1.36.3          ggraph_2.1.0
[100] nlme_3.1-157             slam_0.1-50              aplot_0.1.8
[103] DO.db_2.9                xml2_1.3.3               compiler_4.2.1
[106] rstudioapi_0.14          png_0.1-7                treeio_1.20.2
[109] tibble_3.2.1             tweenr_2.0.2             stringi_1.7.12
[112] forcats_1.0.0            lattice_0.20-45          Matrix_1.6-4
[115] commonmark_1.9.0         markdown_1.9             vctrs_0.6.1
[118] pillar_1.9.0             lifecycle_1.0.3          BiocManager_1.30.18
[121] GlobalOptions_0.1.2      simplifyEnrichment_1.6.1 data.table_1.14.4
[124] bitops_1.0-7             patchwork_1.1.2          qvalue_2.28.0
[127] R6_2.5.1                 renv_1.0.3               gridExtra_2.3
[130] IRanges_2.30.1           codetools_0.2-18         MASS_7.3-57
[133] rprojroot_2.0.3          rjson_0.2.21             withr_2.5.0
[136] S4Vectors_0.34.0         GenomeInfoDbData_1.2.8   org.At.tair.db_3.15.1
[139] ggtext_0.1.2             parallel_4.2.1           hms_1.1.3
[142] clusterProfiler_4.4.4    grid_4.2.1               ggfun_0.1.1
[145] tidyr_1.3.0              rmarkdown_2.24           ggforce_0.4.1
[148] Biobase_2.56.0           NLP_0.2-1
```