

Create a csv file to access all data set.

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

Load list created from Araport11 GFF3

```
inf <-
  fs::path(wd, "misc",
            "AGI_description_list_from_araport11gff3.csv")

tbl_AGI_desc_list <-
  inf %>%
  readr::read_csv()
```

```
Rows: 33341 Columns: 7
— Column specification —————
Delimiter: ","
chr (7): AGI, locus_type, symbol, alias, full_name, curator_summary, descrip...

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.
```

```
path_file_ext_rm <- function(x) fs::path_ext_remove(fs::path_file(x))

left_join_w_warning <- function(x, ...) {
  nr <- nrow(x)
  tbl <- dplyr::left_join(x, ...)
  if(nr != nrow(tbl))
    warning("Number of rows has chaged. The key in y might be not unique.")
  return(tbl)
}
```

Combine with read count and TPM data

```
inf <- fs::path(wd, "data_preproc", "readcount", "count_rna_exon", "count_by_gene.csv")
tbl <- readr::read_csv(inf, col_types = "ciiiiiiiiiiiddddddddddd")

tbl %>%
  dplyr::select(
    AGI = Geneid,
    dplyr::starts_with("zt0"), dplyr::starts_with("zt3"),
    dplyr::starts_with("zt6"), dplyr::starts_with("zt12"),
    dplyr::starts_with("zt18"), dplyr::starts_with("zt21")
  ) %>%
  readr::write_csv(path_out("rna_readcount.csv"))

tbl %>%
  dplyr::select(
    AGI = Geneid,
    dplyr::starts_with("tpm_zt0"), dplyr::starts_with("tpm_zt3"),
    dplyr::starts_with("tpm_zt6"), dplyr::starts_with("tpm_zt12"),
    dplyr::starts_with("tpm_zt18"), dplyr::starts_with("tpm_zt21")
  ) %>%
  readr::write_csv(path_out("rna_tpm.csv"))

tbl_AGI_desc_list <-
  tbl %>%
  dplyr::select(AGI = Geneid, !Length) %>%
  left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")

sf_default_rna <- readRDS(fs::path(wd, "analysis", "deseq2_rna", "sf_default_rna.rds"))
tbl <-
  readr::read_csv(inf) %>%
  dplyr::select(!c(Length, dplyr::starts_with("tpm_"))) %>%
  ngsmisc::ds2_tbl_to_rcdf() %>%
  tibble::as_tibble(rownames = "AGI") %>%
  purrr::imodify(~ {
    if(any(.y == names(sf_default_rna))) {
      return(.x / sf_default_rna[.y])
    } else {
      return(.x)
    }
  }) %>%
  purrr::set_names(~ ifelse(.x == "AGI", .x, paste0("nc_", .x)))
```

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.

```
tbl %>%
  dplyr::select(
    AGI,
    dplyr::starts_with("nc_zt0"), dplyr::starts_with("nc_zt3"),
    dplyr::starts_with("nc_zt6"), dplyr::starts_with("nc_zt12"),
    dplyr::starts_with("nc_zt18"), dplyr::starts_with("nc_zt21")
  ) %>%
  readr::write_csv(path_out("rna_nc.csv"))
```

```
tbl_AGI_desc_list <-
  tbl %>%
  left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")

inf <- fs::path(wd, "data_preproc", "readcount", "count_ribo_central_cds_psites",
"count_by_gene.csv")
tbl <-
  readr::read_csv(inf, col_types = "ciiiiiiiiiiiddddddddddd") %>%
  dplyr::select(AGI = Geneid, !Length) %>%
  purrr::set_names(~ stringr::str_replace(.x, "_ribo", "_morf"))

tbl %>%
  dplyr::select(
    AGI,
    dplyr::starts_with("zt0"), dplyr::starts_with("zt3"),
    dplyr::starts_with("zt6"), dplyr::starts_with("zt12"),
    dplyr::starts_with("zt18"), dplyr::starts_with("zt21")
  ) %>%
  readr::write_csv(path_out("morf_readcount.csv"))

tbl %>%
  dplyr::select(
    AGI,
    dplyr::starts_with("tpm_zt0"), dplyr::starts_with("tpm_zt3"),
    dplyr::starts_with("tpm_zt6"), dplyr::starts_with("tpm_zt12"),
    dplyr::starts_with("tpm_zt18"), dplyr::starts_with("tpm_zt21")
  ) %>%
  readr::write_csv(path_out("morf_tpm.csv"))

tbl_AGI_desc_list <-
  tbl %>%
  left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")

sf_default_ribo <- readRDS(fs::path(wd, "analysis", "deseq2_ribo", "sf_default_ribo.rds"))
tbl <-
  readr::read_csv(inf) %>%
  dplyr::select(!c(Length, dplyr::starts_with("tpm_"))) %>%
  ngsmisc::ds2_tbl_to_rcdf() %>%
  tibble::as_tibble(rownames = "AGI") %>%
  purrr::imodify(~ {
    if(any(.y == names(sf_default_ribo))) {
      return(.x / sf_default_ribo[.y])
    } else {
      return(.x)
    }
  }) %>%
  purrr::set_names(~ stringr::str_replace(.x, "_ribo", "_morf")) %>%
  purrr::set_names(~ ifelse(.x == "AGI", .x, paste0("nc_", .x)))
```

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.

```
tbl %>%
  dplyr::select(
    AGI,
    dplyr::starts_with("nc_zt0"), dplyr::starts_with("nc_zt3"),
    dplyr::starts_with("nc_zt6"), dplyr::starts_with("nc_zt12"),
    dplyr::starts_with("nc_zt18"), dplyr::starts_with("nc_zt21")
  ) %>%
  readr::write_csv(path_out("morf_nc.csv"))

tbl_AGI_desc_list <-
  tbl %>%
  left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")
```

```
inf <- fs::path(wd, "data_preproc", "readcount", "count_ribo_uorf_psites", "count_by_gene.csv")
tbl <-
  readr::read_csv(inf, col_types = "ciiiiiiiiiiiddddddddddd") %>%
  dplyr::select(AGI = Geneid, !Length) %>%
  purrr::set_names(~ stringr::str_replace(.x, "_ribo", "_uorf"))

tbl %>%
  dplyr::select(
    AGI,
    dplyr::starts_with("zt0"), dplyr::starts_with("zt3"),
    dplyr::starts_with("zt6"), dplyr::starts_with("zt12"),
    dplyr::starts_with("zt18"), dplyr::starts_with("zt21")
  ) %>%
  readr::write_csv(path_out("uorf_readcount.csv"))

tbl %>%
  dplyr::select(
    AGI,
    dplyr::starts_with("tpm_zt0"), dplyr::starts_with("tpm_zt3"),
    dplyr::starts_with("tpm_zt6"), dplyr::starts_with("tpm_zt12"),
    dplyr::starts_with("tpm_zt18"), dplyr::starts_with("tpm_zt21")
  ) %>%
  readr::write_csv(path_out("uorf_tpm.csv"))

tbl_AGI_desc_list <-
  tbl %>%
  left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")

sf_default_ribo <- readRDS(fs::path(wd, "analysis", "deseq2_ribo", "sf_default_ribo.rds"))
tbl <-
  readr::read_csv(inf) %>%
  dplyr::select(!c(Length, dplyr::starts_with("tpm_"))) %>%
  ngsmisc::ds2_tbl_to_rcdf() %>%
  tibble::as_tibble(rownames = "AGI") %>%
  purrr::imodify(~ {
    if(any(.y == names(sf_default_ribo))) {
      return(.x / sf_default_ribo[.y])
    } else {
      return(.x)
    }
  }) %>%
  purrr::set_names(~ stringr::str_replace(.x, "_ribo", "_uorf")) %>%
  purrr::set_names(~ ifelse(.x == "AGI", .x, paste0("nc_", .x)))
```

Rows: 4778 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.

```
tbl %>%
  dplyr::select(
    AGI,
    dplyr::starts_with("nc_zt0"), dplyr::starts_with("nc_zt3"),
    dplyr::starts_with("nc_zt6"), dplyr::starts_with("nc_zt12"),
    dplyr::starts_with("nc_zt18"), dplyr::starts_with("nc_zt21")
  ) %>%
  readr::write_csv(path_out("uorf_nc.csv"))

tbl_AGI_desc_list <-
  tbl %>%
  left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")
```

DESeq2 rna

```
tbl_AGI_desc_list <-
  fs::path(wd, "analysis", "deseq2_rna", "deg_all.csv") %>%
  readr::read_csv(col_types = "cddliddilddddddddddllidldd") %>%
  purrr::set_names(nm = function(x){
    x[-1] <- paste0("rna_", x[-1])
    x
  }) %>%
  left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")
```

DESeq2 ribo

```
tbl_AGI_desc_list <-
  fs::path(wd, "analysis", "deseq2_ribo", "deg_all.csv") %>%
  readr::read_csv(col_types = "cddliddilddddddddddllidldd") %>%
  purrr::set_names(nm = function(x){
    x[-1] <- paste0("morf_", x[-1])
    x
  }) %>%
  left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")
```

DESeq2 TE

```
tbl_AGI_desc_list <-
  fs::path(wd, "analysis", "deseq2_te", "deg_all.csv") %>%
  readr::read_csv(col_types = "cddliddilddddddddddllidldd") %>%
  purrr::set_names(nm = function(x){
    x[-1] <- paste0("te_morf_", x[-1])
    x
  }) %>%
  left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")
```

DESeq2 ribo uorf

```
tbl_AGI_desc_list <-
  fs::path(wd, "analysis", "deseq2_ribo_uorf", "deg_all.csv") %>%
  readr::read_csv(col_types = "cddliddilddddddddddllidldd") %>%
  purrr::set_names(nm = function(x){
```

```
x[-1] <- paste0("uorf_", x[-1])
x
}) %>%
left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")
```

DESeq2 TE uorf

```
tbl_AGI_desc_list <-
fs::path(wd, "analysis", "deseq2_te_uorf", "deg_all.csv") %>%
readr::read_csv(col_types = "cddliddildddddlddddldlidd") %>%
purrr::set_names(nm = function(x){
  x[-1] <- paste0("te_uorf_", x[-1])
  x
}) %>%
left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")
```

Combine with eJTK results

```
tbl_AGI_desc_list <-
fs::path(wd, "analysis", "phase_analysis_ejtk", "rna_exon", "data_rna_exon_jtkout_GammaP.txt") %>%
readr::read_tsv() %>%
purrr::set_names(nm = function(x) paste0("rna_", x)) %>%
dplyr::rename(AGI = rna_ID) %>%
dplyr::mutate(rna_BF_BH = p.adjust(rna_BF, "BH"), .after = rna_BF) %>%
left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")
```

Rows: 23981 Columns: 20

— Column specification —

Delimiter: "\t"

chr (2): ID, Waveform

dbl (18): Period, Phase, Nadir, Mean, Std_Dev, MaxLoc, MinLoc, Max, Min, Max...

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.

```
tbl_AGI_desc_list <-
fs::path(wd, "analysis", "phase_analysis_ejtk", "ribo_ccds_psite",
"data_ribo_ccds_psite_jtkout_GammaP.txt") %>%
readr::read_tsv() %>%
purrr::set_names(nm = function(x) paste0("morf_", x)) %>%
dplyr::rename(AGI = morf_ID) %>%
dplyr::mutate(morf_BF_BH = p.adjust(morf_BF, "BH"), .after = morf_BF) %>%
left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")
```

Rows: 21274 Columns: 20

— Column specification —

Delimiter: "\t"

chr (2): ID, Waveform

dbl (18): Period, Phase, Nadir, Mean, Std_Dev, MaxLoc, MinLoc, Max, Min, Max...

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.

```
tbl_AGI_desc_list <-
fs::path(wd, "analysis", "phase_analysis_ejtk", "ribo_uorf_psite",
"data_ribo_uorf_psite_jtkout_GammaP.txt") %>%
readr::read_tsv() %>%
```

```

purrr::set_names(nm = function(x) paste0("uorf_", x)) %>%
dplyr::rename(AGI = uorf_ID) %>%
dplyr::mutate(uorf_BF_BH = p.adjust(uorf_BF, "BH"), .after = uorf_BF) %>%
left_join_w_warning(tbl_AGI_desc_list, y = ., by = "AGI")

```

Rows: 4753 Columns: 20

— Column specification —

Delimiter: "\t"

chr (2): ID, Waveform

dbl (18): Period, Phase, Nadir, Mean, Std_Dev, MaxLoc, MinLoc, Max, Min, Max...

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.

Write to a CSV file

```

readr::write_csv(
  x = tbl_AGI_desc_list,
  file = path_out("summary_all.csv"),
  na = ""
)

```

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] compiler_4.2.1	pillar_1.9.0	BiocManager_1.30.18
[4] tools_4.2.1	bit_4.0.5	digest_0.6.31
[7] jsonlite_1.8.4	evaluate_0.20	lifecycle_1.0.3
[10] tibble_3.2.1	gtable_0.3.1	pkgconfig_2.0.3
[13] rlang_1.1.0	cli_3.6.0	rstudioapi_0.14
[16] parallel_4.2.1	yaml_2.3.6	xfun_0.40
[19] fastmap_1.1.0	stringr_1.5.0	withr_2.5.0
[22] dplyr_1.1.1	knitr_1.42	ngsmisc_0.4.0
[25] generics_0.1.3	fs_1.5.2	vctrs_0.6.1
[28] hms_1.1.3	bit64_4.0.5	rprojroot_2.0.3
[31] grid_4.2.1	tidyselect_1.2.0	glue_1.6.2
[34] here_1.0.1	R6_2.5.1	fansi_1.0.3
[37] vroom_1.6.0	rmarkdown_2.24	purrr_1.0.1
[40] tzdb_0.3.0	readr_2.1.4	scales_1.2.1

```
[43] htmltools_0.5.3    colorspace_2.0-3    renv_1.0.3  
[46] utf8_1.2.2         stringi_1.7.12      munsell_0.5.0  
[49] crayon_1.5.2
```