Create activelly translated uORF annotation file

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"))
```

Directroy setting

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

Loading data

```
# load ribotricer uORF data from excel file
inf <- fs::path(wd, "analysis", "uorf_data", "araport11_uorf_ribotricer.xlsx")
readxl::excel_sheets(inf)
```

```
[1] "uORF containing genes"       "uORF containing transcripts"
[3] "uORF position"               "uORF position id"
```

```
tbl_uorf_pos <- readxl::read_excel(inf, sheet = 3)
tbl_uorf_id <- readxl::read_excel(inf, sheet = 4)
```

Extract unique uORF ID

```
unique_uorf_id <-
  tbl_uorf_id %>%
  dplyr::arrange(uorf_id) %>%
  dplyr::with_groups(id, dplyr::slice_head, n = 1)
```

Extract uORF ID which overlapped with main ORFs

```
# Arabidopsis Genome DNA sequence
bsg_tair <- BSgenome::getBSgenome("BSgenome.Athaliana.TAIR.TAIR9")
```

```
```

```
Attaching package: 'BiocGenerics'
```

```
The following objects are masked from 'package:stats':

    IQR, mad, sd, var, xtabs
```

```
The following objects are masked from 'package:base':

    anyDuplicated, append, as.data.frame, basename, cbind, colnames,
    dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
    grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
    union, unique, unsplit, which.max, which.min
```

```
Attaching package: 'S4Vectors'
```

```
The following objects are masked from 'package:base':

    expand.grid, I, unname
```

```
Attaching package: 'GenomicRanges'
```

```
The following object is masked from 'package:magrittr':

    subtract
```

```
Attaching package: 'Biostrings'
```

```
The following object is masked from 'package:base':

    strsplit
```

```
# Genomic annotation info from Araport11
txdb_araport <-
  GenomicFeatures::makeTxDbFromGFF(
    file = fs::path(wd, "misc", "gff_gtf",
                    "Araport11_GFF3_genes_transposons.201606_mod.gff"),
    format = "gff3",
    dataSource = "Araport11",
    organism = "Arabidopsis thaliana",
    circ_seqs = c("ChrC", "ChrM"),
    chrominfo = GenomeInfoDb::seqinfo(bsg_tair)
  )
```

```
Import genomic features from the file as a GRanges object ...
```

```
OK
```

```
Prepare the 'metadata' data frame ... OK
Make the TxDb object ...
```

```
Warning in .get_cds_IDX(mcols0$type, mcols0$phase): The "phase" metadata column contains non-NA
values for features of type
  exon. This information was ignored.
```

```
OK
```

```
# Find uORFs overlapped with mORFs
uorf_id_overlapped <-
  tbl_uorf_pos %>%
  plyranges::as_granges() %>%
  plyranges::find_overlaps_directed(GenomicFeatures::cds(txdb_araport)) %>%
  tibble::as_tibble() %>%
  dplyr::pull(uorf_id) %>%
  unique()
str(uorf_id_overlapped)
```

```
 chr [1:17037] "AT1G01020.3_8345_8666_216" "AT1G01020.6_8419_8442_24" ...
```

```
# Load P-site count data
tbl_count <-
  fs::path(wd, "data_preproc", "readcount",
           "count_ribo_uorf_psite_all", "count_by_gene.csv") %>%
  readr::read_csv(show_col_types = FALSE) %>%
  dplyr::select(Geneid, Length, dplyr::matches("^zt"))

# Calculate normalized P-site count using scale factors from DESeq2
sf_default_ribo <-
  fs::path(wd, "analysis", "deseq2_ribo", "sf_default_ribo.rds") %>%
  readRDS()
tbl_norm_count <-
  tbl_count %>%
  purrr::imodify(~ {
    if(any(names(sf_default_ribo) %in% .y)) .x / sf_default_ribo[.y]
    else .x
  }) %>%
  dplyr::select(!Length)

tbl_average <-
  tbl_norm_count %>%
  dplyr::rename(uorf_id = Geneid) %>%
  dplyr::mutate(first_uorf_id = stringr::str_extract(uorf_id, pattern = "^[^,]+")) %>%
  dplyr::filter(!(first_uorf_id %in% uorf_id_overlapped)) %>%
  dplyr::select(!first_uorf_id) %>%
  dplyr::mutate(
    zt_0_read = (zt0_1_ribo + zt0_2_ribo) / 2,
    zt_3_read = (zt3_1_ribo + zt3_2_ribo) / 2,
    zt_6_read = (zt6_1_ribo + zt6_2_ribo) / 2,
    zt_12_read = (zt12_1_ribo + zt12_2_ribo) / 2,
    zt_18_read = (zt18_1_ribo + zt18_2_ribo) / 2,
    zt_21_read = (zt21_1_ribo + zt21_2_ribo) / 2
  ) %>%
  dplyr::select(uorf_id, dplyr::matches("_read$"))

tbl_average_per_codon <-
  tbl_average %>%
```

```
  dplyr::left_join(dplyr::select(tbl_count, uorf_id = Geneid, len = Length), by = "uorf_id") %>%
  dplyr::mutate(
    zt_0_read = zt_0_read / (len / 3),
    zt_3_read = zt_3_read / (len / 3),
    zt_6_read = zt_6_read / (len / 3),
    zt_12_read = zt_12_read / (len / 3),
    zt_18_read = zt_18_read / (len / 3),
    zt_21_read = zt_21_read / (len / 3)
  )
```

Write actively translated uORF data to annotation file

```
temp <-
  tbl_average_per_codon %>%
  purrr::modify_at(~ grepl("read", .x), function(x) x >= 0.5) %>%
  dplyr::mutate(any = zt_0_read | zt_3_read | zt_6_read |
                  zt_12_read | zt_18_read | zt_21_read) %>%
  dplyr::filter(any)
temp$uorf_id %>% stringr::str_sub(1, 9) %>% unique() %>% length()
```

```
[1] 4778
```

```
temp <-
  temp %>%
  dplyr::rowwise() %>%
  dplyr::mutate(uorf_id = stringr::str_split(uorf_id, ",")) %>%
  dplyr::ungroup() %>%
  tidyr::unnest_longer(uorf_id)

tbl_uorfs <-
  tbl_uorf_pos %>%
  dplyr::filter(uorf_id %in% temp$uorf_id) %>%
  dplyr::arrange(seqnames, start, end, transcript_id)

# the coordinates for each uORF as an ID of each uORF
tbl_uorfs_2 <-
  tbl_uorfs %>%
  dplyr::left_join(tbl_uorf_id, by = "uorf_id") %>%
  dplyr::group_by(uorf_id, id) %>%
  tidyr::nest()

# uORFs with matching coordinates are merged
tbl_uorfs_3 <-
  tbl_uorfs_2 %>%
  dplyr::group_by(id) %>%
  dplyr::mutate(name = paste0(uorf_id, collapse = ",")) %>%
  dplyr::ungroup() %>%
  dplyr::select(name, data) %>%
  tidyr::unnest() %>%
  dplyr::select(-c(gene_id, transcript_id, uorf_type)) %>%
  dplyr::distinct()
```

```
Warning: `cols` is now required when using `unnest()`.
ℹ Please use `cols = c(data)`.
```

```
outf <- fs::path(wd, "data_modified", "gff_gtf", "araport11_active_uorf_ribotricer.gff3")
readr::write_lines(
  c(
```

```
    "##gff-version 3",
    paste0("##date ", lubridate::today())
  ),
  outf
)

tbl_uorfs_3 %>%
  dplyr::mutate(source = "ribotricer", feature = "uORF", score = ".", frame = ".") %>%
  dplyr::mutate(gene_id = stringr::str_sub(name, 1, 9)) %>%
  dplyr::mutate(attributes = stringr::str_glue('ID="{name}"; gene_id="{gene_id}"')) %>%
  dplyr::select(seqnames, source, feature, start, end,
                score, strand, frame, attributes) %>%
  write.table(outf, quote = FALSE, row.names = FALSE, col.names = FALSE,
              sep = "\t", append = TRUE)
```

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] stats4    stats     graphics  grDevices datasets  utils     methods
[8] base

other attached packages:
 [1] BSgenome.Athaliana.TAIR.TAIR9_1.3.1000
 [2] BSgenome_1.64.0
 [3] rtracklayer_1.56.1
 [4] Biostrings_2.64.1
 [5] XVector_0.36.0
 [6] GenomicRanges_1.48.0
 [7] GenomeInfoDb_1.32.4
 [8] IRanges_2.30.1
 [9] S4Vectors_0.34.0
[10] BiocGenerics_0.42.0
[11] ggplot2_3.4.2
[12] magrittr_2.0.3

loaded via a namespace (and not attached):
 [1] bitops_1.0-7                matrixStats_0.62.0
 [3] fs_1.5.2                    lubridate_1.9.2
 [5] bit64_4.0.5                 filelock_1.0.2
 [7] progress_1.2.2              httr_1.4.5
 [9] rprojroot_2.0.3             tools_4.2.1
[11] utf8_1.2.2                  R6_2.5.1
[13] DBI_1.1.3                   colorspace_2.0-3
[15] withr_2.5.0                 tidyselect_1.2.0
[17] prettyunits_1.1.1           bit_4.0.5
[19] curl_4.3.3                  compiler_4.2.1
[21] cli_3.6.0                   Biobase_2.56.0
```

```
[23] xml2_1.3.3                 DelayedArray_0.22.0
[25] scales_1.2.1               readr_2.1.4
[27] rappdirs_0.3.3             stringr_1.5.0
[29] digest_0.6.31              Rsamtools_2.12.0
[31] rmarkdown_2.24             pkgconfig_2.0.3
[33] htmltools_0.5.3            MatrixGenerics_1.8.1
[35] dbplyr_2.3.2               fastmap_1.1.0
[37] rlang_1.1.0                readxl_1.4.2
[39] rstudioapi_0.14            RSQLite_2.2.18
[41] BiocIO_1.6.0               generics_0.1.3
[43] jsonlite_1.8.4             vroom_1.6.0
[45] BiocParallel_1.30.4        dplyr_1.1.1
[47] RCurl_1.98-1.9             GenomeInfoDbData_1.2.8
[49] Matrix_1.6-4               Rcpp_1.0.11
[51] munsell_0.5.0              fansi_1.0.3
[53] lifecycle_1.0.3            stringi_1.7.12
[55] yaml_2.3.6                 SummarizedExperiment_1.26.1
[57] zlibbioc_1.42.0            BiocFileCache_2.4.0
[59] grid_4.2.1                 blob_1.2.3
[61] parallel_4.2.1             crayon_1.5.2
[63] lattice_0.20-45            GenomicFeatures_1.48.4
[65] hms_1.1.3                  KEGGREST_1.36.3
[67] knitr_1.42                 pillar_1.9.0
[69] rjson_0.2.21               codetools_0.2-18
[71] biomaRt_2.52.0             XML_3.99-0.11
[73] glue_1.6.2                 evaluate_0.20
[75] renv_1.0.3                 BiocManager_1.30.18
[77] vctrs_0.6.1                png_0.1-7
[79] tzdb_0.3.0                 cellranger_1.1.0
[81] tidyr_1.3.0                purrr_1.0.1
[83] gtable_0.3.1               cachem_1.0.6
[85] xfun_0.40                  restfulr_0.0.15
[87] tibble_3.2.1               GenomicAlignments_1.32.1
[89] AnnotationDbi_1.58.0       plyranges_1.16.0
[91] memoise_2.0.1              timechange_0.1.1
[93] here_1.0.1
```