Create uORF annotation file from ribotricer index 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"))
```

Load reference sequences and annotations

```
# 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
```

Load uORF data

```
# Load the candidate ORF index data from the ribotricer
tbl_uorf_info <-
  fs::path(wd, "data_preproc", "ribotricer_out", "araport11_candidate_orfs.tsv") %>%
  readr::read_tsv(show_col_types = FALSE)
tbl_uorf_info %>% dplyr::glimpse()
```

```
Rows: 473,362
Columns: 11
$ ORF_ID          <chr> "AT1G01010.1_3760_5627_1287", "AT1G01020.1_6918_8666_7…
$ ORF_type        <chr> "annotated", "annotated", "annotated", "annotated", "a…
$ transcript_id   <chr> "AT1G01010.1", "AT1G01020.1", "AT1G01020.3", "AT1G0102…
$ transcript_type <chr> "assumed_protein_coding", "assumed_protein_coding", "a…
$ gene_id         <chr> "AT1G01010", "AT1G01020", "AT1G01020", "AT1G01020", "A…
$ gene_name       <chr> "AT1G01010", "AT1G01020", "AT1G01020", "AT1G01020", "A…
$ gene_type       <chr> "assumed_protein_coding", "assumed_protein_coding", "a…
$ chrom           <chr> "Chr1", "Chr1", "Chr1", "Chr1", "Chr1", "Chr1", "Chr1"…
$ strand          <chr> "+", "-", "-", "-", "-", "-", "-", "-", "-", "+", "+",…
$ start_codon     <chr> "ATG", "ATG", "ATG", "ATG", "ATG", "ATG", "ATG", "ATG"…
$ coordinate      <chr> "3760-3913,3996-4276,4486-4605,4706-5095,5174-5326,543…
```

```
tbl_uorf_info$ORF_type %>% table()
```

```
.
   annotated          dORF         novel overlap_dORF overlap_uORF   super_dORF
       48358         23886        143438        24172         7357       150884
  super_uORF          uORF
       58075         17192
```

```
# Filter rows which have the ORF_type column value are uORF related.
tbl_uorf_info <-
  tbl_uorf_info %>%
  dplyr::filter(ORF_type %in% c("overlap_uORF", "super_uORF", "uORF"))
tbl_uorf_info %>% dplyr::glimpse()
```

```
Rows: 82,624
Columns: 11
$ ORF_ID          <chr> "AT1G01020.6_8629_8646_18", "AT1G01020.6_8419_8442_24"…
$ ORF_type        <chr> "uORF", "overlap_uORF", "uORF", "super_uORF", "super_u…
$ transcript_id   <chr> "AT1G01020.6", "AT1G01020.6", "AT1G01020.6", "AT1G0102…
$ transcript_type <chr> "assumed_protein_coding", "assumed_protein_coding", "a…
$ gene_id         <chr> "AT1G01020", "AT1G01020", "AT1G01020", "AT1G01020", "A…
$ gene_name       <chr> "AT1G01020", "AT1G01020", "AT1G01020", "AT1G01020", "A…
$ gene_type       <chr> "assumed_protein_coding", "assumed_protein_coding", "a…
$ chrom           <chr> "Chr1", "Chr1", "Chr1", "Chr1", "Chr1", "Chr1", "Chr1"…
$ strand          <chr> "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-",…
$ start_codon     <chr> "ATG", "ATG", "ATG", "ATG", "ATG", "ATG", "ATG", "ATG"…
$ coordinate      <chr> "8629-8646", "8419-8442", "8442-8464,8594-8666", "9077…
```

uORF data pre-processing

```
# Create a tibble containing uORF coordinate information
tbl_uorf_pos <-
  tbl_uorf_info %>%
  tidyr::separate_rows(coordinate, sep = ",") %>%
  tidyr::separate(coordinate, c("start", "end"), sep = "-", convert = TRUE) %>%
  dplyr::select(transcript_id, gene_id, seqnames = chrom, start, end, strand,
              uorf_id = ORF_ID, uorf_type = ORF_type) %>%
  dplyr::mutate(width = end - start + 1L, .after = end) %>%
  dplyr::arrange(seqnames, start)
tbl_uorf_pos
```

```
# A tibble: 89,573 × 9
   transcript_id gene_id   seqnames start   end width strand uorf_id   uorf_type
   <chr>         <chr>     <chr>    <int> <int> <int> <chr>  <chr>     <chr>
 1 AT1G01020.3   AT1G01020 Chr1      8345  8464   120 -      AT1G0102… overlap_…
 2 AT1G01020.6   AT1G01020 Chr1      8419  8442    24 -      AT1G0102… overlap_…
 3 AT1G01020.5   AT1G01020 Chr1      8419  8442    24 -      AT1G0102… overlap_…
 4 AT1G01020.6   AT1G01020 Chr1      8442  8464    23 -      AT1G0102… uORF
 5 AT1G01020.4   AT1G01020 Chr1      8442  8464    23 -      AT1G0102… overlap_…
 6 AT1G01020.5   AT1G01020 Chr1      8442  8464    23 -      AT1G0102… uORF
 7 AT1G01020.3   AT1G01020 Chr1      8442  8464    23 -      AT1G0102… overlap_…
 8 AT1G01020.3   AT1G01020 Chr1      8571  8574     4 -      AT1G0102… overlap_…
 9 AT1G01020.3   AT1G01020 Chr1      8571  8666    96 -      AT1G0102… overlap_…
10 AT1G01020.6   AT1G01020 Chr1      8594  8666    73 -      AT1G0102… uORF
# ℹ 89,563 more rows
```

```
dir_output <- fs::path("analysis", "uorf_data")
fs::dir_create(dir_output)
readr::write_csv(tbl_uorf_pos, fs::path(wd, dir_output, "tbl_uorf_pos.csv"))

# Create a tibble containing uorf_id and id (unique position identifier)
tbl_uorf_id <-
  tbl_uorf_pos %>%
  dplyr::group_by(uorf_id) %>%
  tidyr::nest() %>%
  dplyr::group_split() %>%
  purrr::map(.f = function(df) {
    dplyr::mutate(df, id =
                    dplyr::select(df$data[[1]], c(3:5, 7)) %>%
                    unlist(recursive = TRUE) %>%
                    paste0(collapse = " "))
  }) %>%
  dplyr::bind_rows() %>%
  dplyr::select(uorf_id, id)
tbl_uorf_id
```

```
# A tibble: 82,624 × 2
   uorf_id                   id
   <chr>                     <chr>
 1 AT1G01020.1_8758_8772_15  Chr1 8758 8772 -
 2 AT1G01020.1_8827_8925_99  Chr1 8827 8925 -
 3 AT1G01020.1_8891_8920_30  Chr1 8891 8920 -
 4 AT1G01020.1_8901_8945_45  Chr1 8901 8945 -
 5 AT1G01020.1_8945_8956_12  Chr1 8945 8956 -
 6 AT1G01020.1_8970_8984_15  Chr1 8970 8984 -
 7 AT1G01020.1_9077_9088_12  Chr1 9077 9088 -
 8 AT1G01020.3_8345_8666_216 Chr1 Chr1 8345 8571 8464 8666 - -
 9 AT1G01020.3_8442_8574_27  Chr1 Chr1 8442 8571 8464 8574 - -
10 AT1G01020.3_8629_8646_18  Chr1 8629 8646 -
# ℹ 82,614 more rows
```

```
readr::write_csv(tbl_uorf_id, fs::path(wd, dir_output, "tbl_uorf_id.csv"))
```

Write uORF data to the annotation file

```
tbl_uorfs <-
  tbl_uorf_pos %>%
  dplyr::arrange(seqnames, start, end, transcript_id)

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

# Merge uORFs share the identical coordinate
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(cols = c(data)) %>%
  dplyr::select(-c(gene_id, transcript_id, uorf_type)) %>%
  dplyr::distinct()

# Write out uORF data to the GFF3 file
outf <- fs::path(wd, "data_modified", "gff_gtf", "araport11_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(attributes = stringr::str_glue('ID="{name}";')) %>%
  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                 rstudioapi_0.14
[39] RSQLite_2.2.18              BiocIO_1.6.0
[41] generics_0.1.3              jsonlite_1.8.4
[43] BiocParallel_1.30.4         vroom_1.6.0
[45] dplyr_1.1.1                 RCurl_1.98-1.9
[47] GenomeInfoDbData_1.2.8      Matrix_1.6-4
[49] Rcpp_1.0.11                 munsell_0.5.0
[51] fansi_1.0.3                 lifecycle_1.0.3
[53] stringi_1.7.12              yaml_2.3.6
[55] SummarizedExperiment_1.26.1 zlibbioc_1.42.0
[57] BiocFileCache_2.4.0         grid_4.2.1
[59] blob_1.2.3                  parallel_4.2.1
[61] crayon_1.5.2                lattice_0.20-45
[63] GenomicFeatures_1.48.4      hms_1.1.3
[65] KEGGREST_1.36.3             knitr_1.42
[67] pillar_1.9.0                rjson_0.2.21
[69] codetools_0.2-18            biomaRt_2.52.0
[71] XML_3.99-0.11               glue_1.6.2
[73] evaluate_0.20               renv_1.0.3
[75] BiocManager_1.30.18         png_0.1-7
[77] vctrs_0.6.1                 tzdb_0.3.0
[79] gtable_0.3.1                purrr_1.0.1
[81] tidyr_1.3.0                 cachem_1.0.6
[83] xfun_0.40                   restfulr_0.0.15
[85] tibble_3.2.1                GenomicAlignments_1.32.1
[87] AnnotationDbi_1.58.0        memoise_2.0.1
[89] timechange_0.1.1            here_1.0.1
```