```
#!/usr/bin/env Rscript

###############################################################################
# This file is part of polap.
#
# polap is free software: you can redistribute it and/or modify it under the
# terms of the GNU General Public License as published by the Free Software
# Foundation, either version 3 of the License, or (at your option) any later
# version.
#
# polap is distributed in the hope that it will be useful, but WITHOUT ANY
# WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
# A PARTICULAR PURPOSE. See the GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License along with
# polap. If not, see <https://www.gnu.org/licenses/>.
###############################################################################

###############################################################################
# This script selects reads using some of minimap2 output files.
# We have multiple scripts that selects reads.
# This is the 5th one.
#
# 1. run-polap-pairs.R -> the first version used in v0.2.6
# 2. run-polap-r-pairs.R -> used in test-reads in oga
# 3. run-polap-r-bridge.R -> used in test-reads in oga
# 4. run-polap-r-select-reads-polap.R -> the 2nd version not used
# 5. run-polap-r-select-reads-ptgaul.R -> a slim version of the 2nd used in dga
# 6. run-polap-r-directional.R -> used by an older version of dga
#    or _run_polap_original-directional-reads
#
# This used to be used in dga and it is not used in the latest dga.
#
# See Also:
# run-polap-pairs.R
# run-polap-r-pairs.R
# run-polap-r-bridge.R
# run-polap-r-select-reads-polap.R
# run-polap-r-select-reads-ptgaul.R
# run-polap-r-directional.R
# run-polap-function-oga.sh
# run-polap-function-dga.sh
#
# NOTE: move with directional module.
###############################################################################

# name: select long reads
#
# synopsis:
#   run-polap-pairs.R mt.contig.name-1 contig.tab ${MTSEEDSDIR} $SINGLE_MIN $SINGLE_MIN
#
```

```
52 # requirement: executes Flye
53 # flye --nano-raw "$LR3K" --out-dir "${_arg_outdir}" \
54 #   --threads "${_arg_threads}" \
55 #   --stop-after contigger \
56 #   --asm-coverage 30 \
57 #   --genome-size "$EXPECTED_GENOME_SIZE"
58 #
59 # input:
60 #   1. mt.contig.name-1 - contig or edge numbers
61 #   2. contig.tab - minimap2 output modified
62 #   3. seeds directory for output
63 #   4. pair minimum length V11: 3000 for MT, 1000 for PT
64 #   5. bridge minimum length V7: depends: 3000 or 5000
65 #   6. single minimum length V11: 3000 for MT, 0 or 1000 for PT
66 #
67 # output:
68 #   single.names and <contig1-contig2.name> files in the output directory
69 #
70 # MTSEEDSDIR="${_arg_outdir}"/60-mt-${STEP4}/o${MR}/seeds
71 #   run-polap-pairs.R mt.contig.name-1 contig.tab ${MTSEEDSDIR} $PAIR_MIN $BRIDGE_MIN $SINGLE_MIN
72 #   run-polap-pairs.R mt.contig.name-1 contig.tab ${MTSEEDSDIR} $PAIR_MIN $BRIDGE_MIN
73
74 suppressWarnings(suppressPackageStartupMessages(library("optparse")))
75 suppressWarnings(suppressPackageStartupMessages(library("dplyr")))
76 suppressWarnings(suppressPackageStartupMessages(library("readr")))
77 suppressWarnings(suppressPackageStartupMessages(library("purrr")))
78 suppressWarnings(suppressPackageStartupMessages(library("tidyr")))
79 suppressWarnings(suppressPackageStartupMessages(library("ggplot2")))
80
81 debug ← Sys.getenv("_POLAP_DEBUG", unset = "0")
82
83 parser ← OptionParser()
84 parser ← add_option(parser, c("−t", "−−table"),
85   action = "store",
86   help = "input minimap2 PAF−like tabular format file",
87   metavar = "<FILE>"
88 )
89 parser ← add_option(parser, c("−m", "−−mtcontigname"),
90   action = "store",
91   help = "mt.contig.name−1",
92   metavar = "<FILE>"
93 )
94 parser ← add_option(parser, c("−o", "−−out"),
95   action = "store",
96   help = "output directory"
97 )
98 parser ← add_option(parser, c("−r", "−−pair−min"),
99   type = "integer",
100  default = 3000,
101  help = "Minimum length of pair−mapping alignment",
102  metavar = "number"
```

```
)
parser ← add_option(parser, c("−w", "−−single−min"),
  type = "integer",
  default = 3000,
  help = "Minimum length of single−mapping alignment",
  metavar = "number"
)
parser ← add_option(parser, c("−x", "−−bridge−min"),
  type = "integer",
  default = 3000,
  help = "Minimum length of bridging reads",
  metavar = "number"
)
parser ← add_option(parser, c("−−intra−base−ratio"),
  type = "double",
  default = 0.7,
  help = "Minimum of V10/V11",
  metavar = "number"
)
parser ← add_option(parser, c("−−intra−read−ratio"),
  type = "double",
  default = 0.7,
  help = "Minimum of (V4−V3)/V2",
  metavar = "number"
)
parser ← add_option(parser, c("−−inter−base−ratio"),
  type = "double",
  default = 0.7,
  help = "Minimum of V10/V11",
  metavar = "number"
)
parser ← add_option(parser, c("−−use−strand"),
  action = "store_true",
  default = FALSE, help = "will create ptgaul.names"
)
parser ← add_option(parser, c("−−use−reverse"),
  action = "store_true",
  default = FALSE, help = "will use the reads mapped on the reverse complement"
)
parser ← add_option(parser, c("−−create−ptgaul"),
  action = "store_true",
  default = FALSE, help = "will create ptgaul.names"
)
parser ← add_option(parser, c("−−create−single"),
  action = "store_true",
  default = FALSE, help = "will create single.names"
)
parser ← add_option(parser, c("−−create−pair"),
  action = "store_true",
  default = FALSE, help = "will create pair.names"
)
```

```
parser ← add_option(parser, c("−−create−combined"),
  action = "store_true",
  default = FALSE, help = "will create combined.names"
)
parser ← add_option(parser, c("−−all"),
  action = "store_true",
  default = FALSE, help = "will create all the 4 names files."
)
parser ← add_option(parser, c("−−outptgaul"),
  action = "store",
  default = "ptgaul.names",
  help = "ptgaul output file"
)
parser ← add_option(parser, c("−−outsingle"),
  action = "store",
  default = "single.names",
  help = "single output file"
)
parser ← add_option(parser, c("−−outpair"),
  action = "store",
  default = "pair.names",
  help = "pair output file"
)
parser ← add_option(parser, c("−−outcombined"),
  action = "store",
  default = "combined.names",
  help = "combined output file"
)

args1 ← parse_args(parser)

if (is_null(args1$table)) {
  s ← "bioprojects"
  o ← "PRJNA817235−Canavalia_ensiformis"

  input_dir0 ← file.path(".")
  input1 ← file.path(input_dir0, "contig.tab")
  input2 ← file.path(input_dir0, "mt.contig.name−1")
  output1 ← file.path(input_dir0, "02−reads")

  args1 ← parse_args(parser, args = c(
    "−−table", input1,
    "−−mtcontigname", input2,
    "−o", output1,
    "−r", 10000,
    "−w", 10000,
    "−x", 10000,
    "−−all",
    "−−use−strand",
    "−−use−reverse",
    "−−intra−base−ratio", 0.7,
```

```
      "--intra-read-ratio", 0.7,
      "--inter-base-ratio", 0.7
  ))
}

# mt.contig.name-x
x1 ← as_tibble(read.table(args1$mtcontigname))

# https://lh3.github.io/minimap2/minimap2.html
# Li 2008 - "OUTPUT FORMAT: Minimap2 outputs mapping positions in the
# Pairwise mApping Format (PAF) by default. PAF is a TAB-delimited text format
# with each line consisting of at least 12 fields as are described in the
# following table:
#
# Col Type  Description
# 1 string  Query sequence name
# 2 int Query sequence length
# 3 int Query start coordinate (0-based)
# 4 int Query end coordinate (0-based)
# 5 char  âM-^@M-^X+âM-^@M-^Y if query/target on the same strand; âM-^@M-^X-âM-^@M-^Y if opposite
# 6 string  Target sequence name
# 7 int Target sequence length
# 8 int Target start coordinate on the original strand
# 9 int Target end coordinate on the original strand
# 10  int Number of matching bases in the mapping
# 11  int Number bases, including gaps, in the mapping
# 12  int Mapping quality (0-255 with 255 for missing)
#
# Read in the data with assigned column names
data ← read_tsv(args1$table,
  show_col_types = FALSE,
  col_names = c(
    "rname",
    "rlen",
    "rstart",
    "rend",
    "strand",
    "cname",
    "clen",
    "cstart",
    "cend",
    "match",
    "base"
  )
)

if (args1$`use-strand`) {
  data ← data |>
    filter(strand ≡ "+")
}

```

```
mtdir ← args1$out
pair_min ← as.numeric(args1$`pair-min`)
brigde_min ← as.numeric(args1$`bridge-min`)
single_min ← as.numeric(args1$`single-min`)
a ← as.numeric(args1$`intra-read-ratio`)
b ← as.numeric(args1$`intra-base-ratio`)
c ← as.numeric(args1$`inter-base-ratio`)

ptgaul_option_base ← paste("ptgaul",
  a,
  b,
  single_min,
  c,
  pair_min,
  brigde_min,
  sep = "−"
)
ptgaul_option_base ← paste0(ptgaul_option_base, ".names")
ptgaul_option_base ← "ptgaul.names"

single_option_base ← paste("single",
  a,
  b,
  single_min,
  c,
  pair_min,
  brigde_min,
  sep = "−"
)
single_option_base ← paste0(single_option_base, ".names")

pair_option_base ← paste("pair",
  a,
  b,
  single_min,
  c,
  pair_min,
  brigde_min,
  sep = "−"
)
pair_option_base ← paste0(pair_option_base, ".names")

combined_option_base ← paste("combined",
  a,
  b,
  single_min,
  c,
  pair_min,
  brigde_min,
  sep = "−"
)
```

```
combined_option_base ← paste0(combined_option_base, ".names")

# 0. Check preconditions
# rstart is not greater than rend.
# match is not greater than base.
result ← data %>%
  summarize(all_rows_meet_condition = all(rstart ≤ rend & match ≤ base))

# Check the result
stopifnot(result$all_rows_meet_condition)
stopifnot(pair_min ≥ 0, brigde_min ≥ 0, single_min ≥ 0)

# 1. ptGAUL: we use ptGAUL for the case of single edge reference
if (args1$`create-ptgaul` || args1$all) {
  ptgaul_file ← file.path(mtdir, args1$outptgaul)
  ptgaul_mapped_reads ← data |>
    filter(match / base > args1$`intra-base-ratio`, base > single_min) |>
    select(rname) |>
    distinct(rname)

  ptgaul_mapped_reads |>
    write.table(
      ptgaul_file,
      row.names = FALSE,
      col.names = FALSE,
      quote = FALSE
    )
  ptgaul_option_file ← file.path(mtdir, ptgaul_option_base)
  print(ptgaul_file)
  print(ptgaul_option_file)
  file.copy(ptgaul_file, ptgaul_option_file)
  # file.copy(ptgaul_file, ptgaul_option_file, showWarnings = FALSE)
}

# 2. intra-contig mapping
if (args1$`create-single` || args1$all) {
  single_file ← file.path(mtdir, args1$outsingle)
  intra_contig_mapped_reads ← data |>
    filter(
      (rend - rstart) / rlen > args1$`intra-read-ratio`,
      match / base > args1$`intra-base-ratio`,
      base > single_min
    ) |>
    select(rname) |>
    distinct(rname)

  intra_contig_mapped_reads |>
    write.table(single_file, row.names = FALSE, col.names = FALSE, quote = FALSE)

  single_option_file ← file.path(mtdir, single_option_base)
  print(single_file)
```

```
358   file.copy(single_file, single_option_file)
359   # file.copy(single_file, single_option_file, showWarnings = FALSE)
360 }
361
362 if (nrow(x1) > 1) {
363   y ← t(combn(x1$V1, 2))
364
365   orders ← function(y) {
366     stopifnot(length(y) ≡ 2)
367     z ← as.numeric(gsub("\\D", "", y))
368     bname ← paste0(z[1], "_", z[2])
369     oname ← paste0(mtdir, "/", bname, ".name")
370     y1 ← data |>
371       filter(
372         cname ≡ y[1],
373         match / base > args1$`inter-base-ratio`,
374         base > pair_min,
375         rlen > brigde_min
376       ) |>
377       select(rname) |>
378       distinct(rname)
379     y2 ← data |>
380       filter(
381         cname ≡ y[2],
382         match / base > args1$`inter-base-ratio`,
383         base > pair_min,
384         rlen > brigde_min
385       ) |>
386       select(rname) |>
387       distinct(rname)
388     intersect(y1, y2) |>
389       write.table(oname, row.names = FALSE, col.names = FALSE, quote = FALSE)
390   }
391
392
393   # 3. inter-contig mapping: individual pair
394   if (args1$all) {
395     apply(y, 1, orders)
396   }
397
398   # 4. inter-contig mapping
399   #
400   # Filter data based on the given criteria
401
402   if (args1$`create-pair` || args1$all) {
403     inter_contig_mapped_reads ← data |>
404       # Calculate the match/base ratio
405       mutate(match_base_ratio = match / base) |>
406       # Filter for match/base > 0.7 and base > 3000
407       filter(
408         match_base_ratio > args1$`inter-base-ratio`,
```

```
409           base > pair_min,
410           rlen > brigde_min
411         ) |>
412         # Count the number of contigs for each read
413         group_by(rname) |>
414         # Keep only those reads that are mapped to at least two contigs
415         filter(n_distinct(cname) ≥ 2) |>
416         # Select unique reads
417         distinct(rname) |>
418         ungroup()
419
420       pair_file ← file.path(mtdir, args1$outpair)
421       inter_contig_mapped_reads |>
422         write.table(
423           pair_file,
424           row.names = FALSE,
425           col.names = FALSE,
426           quote = FALSE
427         )
428       pair_option_file ← file.path(mtdir, pair_option_base)
429       print(pair_file)
430       file.copy(pair_file, pair_option_file)
431       # file.copy(pair_file, pair_option_file, showWarnings = FALSE)
432     }
433
434     # 5. combined: single + pair
435     if (args1$`create-combined` || args1$all) {
436       combined_mapped_reads ← bind_rows(
437         inter_contig_mapped_reads,
438         intra_contig_mapped_reads
439       ) |>
440         distinct(rname, .keep_all = TRUE)
441
442       combined_file ← file.path(mtdir, args1$outcombined)
443       combined_mapped_reads |>
444         write.table(
445           combined_file,
446           row.names = FALSE,
447           col.names = FALSE,
448           quote = FALSE
449         )
450       combined_option_file ← file.path(mtdir, combined_option_base)
451       print(combined_file)
452       file.copy(combined_file, combined_option_file)
453       # file.copy(combined_file, combined_option_file, showWarnings = FALSE)
454     }
455   }
```