

7IM-^M-^T 01, 25 10:17

## run-polap-r-bridge.R

Page 1/9

```

1 #!/usr/bin/env Rscript
2
3 ######
4 # This file is part of polap.
5 #
6 # polap is free software: you can redistribute it and/or modify it under the
7 # terms of the GNU General Public License as published by the Free Software
8 # Foundation, either version 3 of the License, or (at your option) any later
9 # version.
10 #
11 # polap is distributed in the hope that it will be useful, but WITHOUT ANY
12 # WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
13 # A PARTICULAR PURPOSE. See the GNU General Public License for more details.
14 #
15 # You should have received a copy of the GNU General Public License along with
16 # polap. If not, see <https://www.gnu.org/licenses/>.
17 #####
18
19 # polaplib/run-polap-r-bridge.R
20 # Check: 2025-06-17
21
22 #####
23 # NOTE: this is the second of the two read-selection scripts.
24 # This script selects reads using some of minimap2 output files.
25 # We have multiple scripts that selects reads.
26 # This is the 3rd one.
27 #
28 # 1. run-polap-pairs.R or polap-r-pairs.R -> the first version used in v0.2.6
29 # 2. run-polap-r-pairs.R -> used in test-reads in oga
30 # 3. run-polap-r-bridge.R -> used in test-reads in oga
31 # 4. run-polap-r-select-reads-polap.R -> the 2nd version not used
32 # 5. run-polap-r-select-reads-ptgaul.R -> a slim version of the 2nd used in dga
33 # 6. run-polap-r-directional.R -> used by an older version of dga
34 #     or _run_polap_original-directional-reads
35 #
36 # Subcommand test-reads uses this script.
37 #
38 # See Also:
39 # run-polap-pairs.R
40 # run-polap-r-pairs.R
41 # run-polap-r-bridge.R
42 # run-polap-r-select-reads-polap.R
43 # run-polap-r-select-reads-ptgaul.R
44 # run-polap-r-directional.R
45 # run-polap-function-oga.sh
46 # run-polap-function-dga.sh
47 #
48 # Check: 2025-06-17
49 #####
50
51 # name: select long reads

```

7|M-[M-^T 01, 25 10:17

**run-polap-r-bridge.R**

Page 2/9

```

52 #
53 # synopsis:
54 #   run-polap-pairs.R mt.contig.name-1 contig.tab ${MTSEEDSDIR} $SINGLE_MIN $SINGLE_MIN
55 #
56 # requirement: executes Flye
57 # flye --nano-raw "$LR3K" --out-dir "${_arg_outdir}" \
58 #   --threads "${_arg_threads}" \
59 #   --stop-after contigger \
60 #   --asm-coverage 30 \
61 #   --genome-size "$EXPECTED_GENOME_SIZE"
62 #
63 # input:
64 #   1. mt.contig.name-1 - contig or edge numbers
65 #   2. contig.tab - minimap2 output modified
66 #   3. seeds directory for output
67 #   4. pair minimum length V11: 3000 for MT, 1000 for PT
68 #   5. bridge minimum length V7: depends: 3000 or 5000
69 #   6. single minimum length V11: 3000 for MT, 0 or 1000 for PT
70 #
71 # output:
72 #   single.names and <contig1-contig2.name> files in the output directory
73 #
74 # MTSEEDSDIR="${_arg_outdir}"/60-mt-${STEP4}/o${MR}/seeds
75 # run-polap-pairs.R mt.contig.name-1 contig.tab ${MTSEEDSDIR} $PAIR_MIN $BRIDGE_MIN $SINGLE_MIN
76 # run-polap-pairs.R mt.contig.name-1 contig.tab ${MTSEEDSDIR} $PAIR_MIN $BRIDGE_MIN
77
78 suppressPackageStartupMessages(library("optparse"))
79 suppressPackageStartupMessages(library("dplyr"))
80 suppressPackageStartupMessages(library("readr"))
81 suppressPackageStartupMessages(library("purrr"))
82 suppressPackageStartupMessages(library("tidyverse"))
83 suppressPackageStartupMessages(library("ggplot2"))
84
85 debug ← Sys.getenv("_POLAP_DEBUG", unset = "0")
86
87 parser ← OptionParser()
88 parser ← add_option(parser, c("-t", "--table"),
89   action = "store",
90   help = "minimap2 PAF",
91   metavar = "<FILE>")
92 )
93 parser ← add_option(parser, c("-m", "--mtcontigname"),
94   action = "store",
95   help = "mt.contig.name-1",
96   metavar = "<FILE>")
97 )
98 parser ← add_option(parser, c("-o", "--out"),
99   action = "store",
100  help = "output directory")
101 )
102 parser ← add_option(parser, c("-r", "--pair-min"),
103   action = "store",
104   help = "pair minimum length")
105 )
106 parser ← add_option(parser, c("-b", "--bridge-min"),
107   action = "store",
108   help = "bridge minimum length")
109 )
110 parser ← add_option(parser, c("-s", "--single-min"),
111   action = "store",
112   help = "single minimum length")
113 )
114 parser ← add_option(parser, c("-c", "--contig-min"),
115   action = "store",
116   help = "contig minimum length")
117 )
118 parser ← add_option(parser, c("-p", "--pair-coverage"),
119   action = "store",
120   help = "pair coverage")
121 )
122 parser ← add_option(parser, c("-g", "--genome-size"),
123   action = "store",
124   help = "genome size")
125 )
126 parser ← add_option(parser, c("-a", "--asm-coverage"),
127   action = "store",
128   help = "asm coverage")
129 )
130 parser ← add_option(parser, c("-n", "--nano-raw"),
131   action = "store",
132   help = "nano raw")
133 )
134 parser ← add_option(parser, c("-t", "--threads"),
135   action = "store",
136   help = "threads")
137 )
138 parser ← add_option(parser, c("-s", "--stop-after"),
139   action = "store",
140   help = "stop after")
141 )
142 parser ← add_option(parser, c("-l", "--log"),
143   action = "store",
144   help = "log")
145 )
146 parser ← add_option(parser, c("-v", "--version"),
147   action = "store",
148   help = "version")
149 )
150 parser ← add_option(parser, c("-h", "--help"),
151   action = "store",
152   help = "help")
153 )
154
155 args ← parse_args(parser)
156
157 if args$version {
158   print_version()
159   exit(0)
160 }
161
162 if args$help {
163   print_usage()
164   exit(0)
165 }
166
167 if args$log {
168   log_file ← file(args$log, "w")
169   log_file ← open(log_file, "w")
170   log_file ← write(log_file, "Log file opened")
171   log_file ← close(log_file)
172 }
173
174 if args$version {
175   print_version()
176   exit(0)
177 }
178
179 if args$help {
180   print_usage()
181   exit(0)
182 }
183
184 if args$version {
185   print_version()
186   exit(0)
187 }
188
189 if args$help {
190   print_usage()
191   exit(0)
192 }
193
194 if args$version {
195   print_version()
196   exit(0)
197 }
198
199 if args$help {
200   print_usage()
201   exit(0)
202 }
203
204 if args$version {
205   print_version()
206   exit(0)
207 }
208
209 if args$help {
210   print_usage()
211   exit(0)
212 }
213
214 if args$version {
215   print_version()
216   exit(0)
217 }
218
219 if args$help {
220   print_usage()
221   exit(0)
222 }
223
224 if args$version {
225   print_version()
226   exit(0)
227 }
228
229 if args$help {
230   print_usage()
231   exit(0)
232 }
233
234 if args$version {
235   print_version()
236   exit(0)
237 }
238
239 if args$help {
240   print_usage()
241   exit(0)
242 }
243
244 if args$version {
245   print_version()
246   exit(0)
247 }
248
249 if args$help {
250   print_usage()
251   exit(0)
252 }
253
254 if args$version {
255   print_version()
256   exit(0)
257 }
258
259 if args$help {
260   print_usage()
261   exit(0)
262 }
263
264 if args$version {
265   print_version()
266   exit(0)
267 }
268
269 if args$help {
270   print_usage()
271   exit(0)
272 }
273
274 if args$version {
275   print_version()
276   exit(0)
277 }
278
279 if args$help {
280   print_usage()
281   exit(0)
282 }
283
284 if args$version {
285   print_version()
286   exit(0)
287 }
288
289 if args$help {
290   print_usage()
291   exit(0)
292 }
293
294 if args$version {
295   print_version()
296   exit(0)
297 }
298
299 if args$help {
300   print_usage()
301   exit(0)
302 }
303
304 if args$version {
305   print_version()
306   exit(0)
307 }
308
309 if args$help {
310   print_usage()
311   exit(0)
312 }
313
314 if args$version {
315   print_version()
316   exit(0)
317 }
318
319 if args$help {
320   print_usage()
321   exit(0)
322 }
323
324 if args$version {
325   print_version()
326   exit(0)
327 }
328
329 if args$help {
330   print_usage()
331   exit(0)
332 }
333
334 if args$version {
335   print_version()
336   exit(0)
337 }
338
339 if args$help {
340   print_usage()
341   exit(0)
342 }
343
344 if args$version {
345   print_version()
346   exit(0)
347 }
348
349 if args$help {
350   print_usage()
351   exit(0)
352 }
353
354 if args$version {
355   print_version()
356   exit(0)
357 }
358
359 if args$help {
360   print_usage()
361   exit(0)
362 }
363
364 if args$version {
365   print_version()
366   exit(0)
367 }
368
369 if args$help {
370   print_usage()
371   exit(0)
372 }
373
374 if args$version {
375   print_version()
376   exit(0)
377 }
378
379 if args$help {
380   print_usage()
381   exit(0)
382 }
383
384 if args$version {
385   print_version()
386   exit(0)
387 }
388
389 if args$help {
390   print_usage()
391   exit(0)
392 }
393
394 if args$version {
395   print_version()
396   exit(0)
397 }
398
399 if args$help {
400   print_usage()
401   exit(0)
402 }
403
404 if args$version {
405   print_version()
406   exit(0)
407 }
408
409 if args$help {
410   print_usage()
411   exit(0)
412 }
413
414 if args$version {
415   print_version()
416   exit(0)
417 }
418
419 if args$help {
420   print_usage()
421   exit(0)
422 }
423
424 if args$version {
425   print_version()
426   exit(0)
427 }
428
429 if args$help {
430   print_usage()
431   exit(0)
432 }
433
434 if args$version {
435   print_version()
436   exit(0)
437 }
438
439 if args$help {
440   print_usage()
441   exit(0)
442 }
443
444 if args$version {
445   print_version()
446   exit(0)
447 }
448
449 if args$help {
450   print_usage()
451   exit(0)
452 }
453
454 if args$version {
455   print_version()
456   exit(0)
457 }
458
459 if args$help {
460   print_usage()
461   exit(0)
462 }
463
464 if args$version {
465   print_version()
466   exit(0)
467 }
468
469 if args$help {
470   print_usage()
471   exit(0)
472 }
473
474 if args$version {
475   print_version()
476   exit(0)
477 }
478
479 if args$help {
480   print_usage()
481   exit(0)
482 }
483
484 if args$version {
485   print_version()
486   exit(0)
487 }
488
489 if args$help {
490   print_usage()
491   exit(0)
492 }
493
494 if args$version {
495   print_version()
496   exit(0)
497 }
498
499 if args$help {
500   print_usage()
501   exit(0)
502 }
503
504 if args$version {
505   print_version()
506   exit(0)
507 }
508
509 if args$help {
510   print_usage()
511   exit(0)
512 }
513
514 if args$version {
515   print_version()
516   exit(0)
517 }
518
519 if args$help {
520   print_usage()
521   exit(0)
522 }
523
524 if args$version {
525   print_version()
526   exit(0)
527 }
528
529 if args$help {
530   print_usage()
531   exit(0)
532 }
533
534 if args$version {
535   print_version()
536   exit(0)
537 }
538
539 if args$help {
540   print_usage()
541   exit(0)
542 }
543
544 if args$version {
545   print_version()
546   exit(0)
547 }
548
549 if args$help {
550   print_usage()
551   exit(0)
552 }
553
554 if args$version {
555   print_version()
556   exit(0)
557 }
558
559 if args$help {
560   print_usage()
561   exit(0)
562 }
563
564 if args$version {
565   print_version()
566   exit(0)
567 }
568
569 if args$help {
570   print_usage()
571   exit(0)
572 }
573
574 if args$version {
575   print_version()
576   exit(0)
577 }
578
579 if args$help {
580   print_usage()
581   exit(0)
582 }
583
584 if args$version {
585   print_version()
586   exit(0)
587 }
588
589 if args$help {
590   print_usage()
591   exit(0)
592 }
593
594 if args$version {
595   print_version()
596   exit(0)
597 }
598
599 if args$help {
600   print_usage()
601   exit(0)
602 }
603
604 if args$version {
605   print_version()
606   exit(0)
607 }
608
609 if args$help {
610   print_usage()
611   exit(0)
612 }
613
614 if args$version {
615   print_version()
616   exit(0)
617 }
618
619 if args$help {
620   print_usage()
621   exit(0)
622 }
623
624 if args$version {
625   print_version()
626   exit(0)
627 }
628
629 if args$help {
630   print_usage()
631   exit(0)
632 }
633
634 if args$version {
635   print_version()
636   exit(0)
637 }
638
639 if args$help {
640   print_usage()
641   exit(0)
642 }
643
644 if args$version {
645   print_version()
646   exit(0)
647 }
648
649 if args$help {
650   print_usage()
651   exit(0)
652 }
653
654 if args$version {
655   print_version()
656   exit(0)
657 }
658
659 if args$help {
660   print_usage()
661   exit(0)
662 }
663
664 if args$version {
665   print_version()
666   exit(0)
667 }
668
669 if args$help {
670   print_usage()
671   exit(0)
672 }
673
674 if args$version {
675   print_version()
676   exit(0)
677 }
678
679 if args$help {
680   print_usage()
681   exit(0)
682 }
683
684 if args$version {
685   print_version()
686   exit(0)
687 }
688
689 if args$help {
690   print_usage()
691   exit(0)
692 }
693
694 if args$version {
695   print_version()
696   exit(0)
697 }
698
699 if args$help {
700   print_usage()
701   exit(0)
702 }
703
704 if args$version {
705   print_version()
706   exit(0)
707 }
708
709 if args$help {
710   print_usage()
711   exit(0)
712 }
713
714 if args$version {
715   print_version()
716   exit(0)
717 }
718
719 if args$help {
720   print_usage()
721   exit(0)
722 }
723
724 if args$version {
725   print_version()
726   exit(0)
727 }
728
729 if args$help {
730   print_usage()
731   exit(0)
732 }
733
734 if args$version {
735   print_version()
736   exit(0)
737 }
738
739 if args$help {
740   print_usage()
741   exit(0)
742 }
743
744 if args$version {
745   print_version()
746   exit(0)
747 }
748
749 if args$help {
750   print_usage()
751   exit(0)
752 }
753
754 if args$version {
755   print_version()
756   exit(0)
757 }
758
759 if args$help {
760   print_usage()
761   exit(0)
762 }
763
764 if args$version {
765   print_version()
766   exit(0)
767 }
768
769 if args$help {
770   print_usage()
771   exit(0)
772 }
773
774 if args$version {
775   print_version()
776   exit(0)
777 }
778
779 if args$help {
780   print_usage()
781   exit(0)
782 }
783
784 if args$version {
785   print_version()
786   exit(0)
787 }
788
789 if args$help {
790   print_usage()
791   exit(0)
792 }
793
794 if args$version {
795   print_version()
796   exit(0)
797 }
798
799 if args$help {
800   print_usage()
801   exit(0)
802 }
803
804 if args$version {
805   print_version()
806   exit(0)
807 }
808
809 if args$help {
810   print_usage()
811   exit(0)
812 }
813
814 if args$version {
815   print_version()
816   exit(0)
817 }
818
819 if args$help {
820   print_usage()
821   exit(0)
822 }
823
824 if args$version {
825   print_version()
826   exit(0)
827 }
828
829 if args$help {
830   print_usage()
831   exit(0)
832 }
833
834 if args$version {
835   print_version()
836   exit(0)
837 }
838
839 if args$help {
840   print_usage()
841   exit(0)
842 }
843
844 if args$version {
845   print_version()
846   exit(0)
847 }
848
849 if args$help {
850   print_usage()
851   exit(0)
852 }
853
854 if args$version {
855   print_version()
856   exit(0)
857 }
858
859 if args$help {
860   print_usage()
861   exit(0)
862 }
863
864 if args$version {
865   print_version()
866   exit(0)
867 }
868
869 if args$help {
870   print_usage()
871   exit(0)
872 }
873
874 if args$version {
875   print_version()
876   exit(0)
877 }
878
879 if args$help {
880   print_usage()
881   exit(0)
882 }
883
884 if args$version {
885   print_version()
886   exit(0)
887 }
888
889 if args$help {
890   print_usage()
891   exit(0)
892 }
893
894 if args$version {
895   print_version()
896   exit(0)
897 }
898
899 if args$help {
900   print_usage()
901   exit(0)
902 }
903
904 if args$version {
905   print_version()
906   exit(0)
907 }
908
909 if args$help {
910   print_usage()
911   exit(0)
912 }
913
914 if args$version {
915   print_version()
916   exit(0)
917 }
918
919 if args$help {
920   print_usage()
921   exit(0)
922 }
923
924 if args$version {
925   print_version()
926   exit(0)
927 }
928
929 if args$help {
930   print_usage()
931   exit(0)
932 }
933
934 if args$version {
935   print_version()
936   exit(0)
937 }
938
939 if args$help {
940   print_usage()
941   exit(0)
942 }
943
944 if args$version {
945   print_version()
946   exit(0)
947 }
948
949 if args$help {
950   print_usage()
951   exit(0)
952 }
953
954 if args$version {
955   print_version()
956   exit(0)
957 }
958
959 if args$help {
960   print_usage()
961   exit(0)
962 }
963
964 if args$version {
965   print_version()
966   exit(0)
967 }
968
969 if args$help {
970   print_usage()
971   exit(0)
972 }
973
974 if args$version {
975   print_version()
976   exit(0)
977 }
978
979 if args$help {
980   print_usage()
981   exit(0)
982 }
983
984 if args$version {
985   print_version()
986   exit(0)
987 }
988
989 if args$help {
990   print_usage()
991   exit(0)
992 }
993
994 if args$version {
995   print_version()
996   exit(0)
997 }
998
999 if args$help {
1000   print_usage()
1001   exit(0)
1002 }
1003
1004 if args$version {
1005   print_version()
1006   exit(0)
1007 }
1008
1009 if args$help {
1010   print_usage()
1011   exit(0)
1012 }
1013
1014 if args$version {
1015   print_version()
1016   exit(0)
1017 }
1018
1019 if args$help {
1020   print_usage()
1021   exit(0)
1022 }
1023
1024 if args$version {
1025   print_version()
1026   exit(0)
1027 }
1028
1029 if args$help {
1030   print_usage()
1031   exit(0)
1032 }
1033
1034 if args$version {
1035   print_version()
1036   exit(0)
1037 }
1038
1039 if args$help {
1040   print_usage()
1041   exit(0)
1042 }
1043
1044 if args$version {
1045   print_version()
1046   exit(0)
1047 }
1048
1049 if args$help {
1050   print_usage()
1051   exit(0)
1052 }
1053
1054 if args$version {
1055   print_version()
1056   exit(0)
1057 }
1058
1059 if args$help {
1060   print_usage()
1061   exit(0)
1062 }
1063
1064 if args$version {
1065   print_version()
1066   exit(0)
1067 }
1068
1069 if args$help {
1070   print_usage()
1071   exit(0)
1072 }
1073
1074 if args$version {
1075   print_version()
1076   exit(0)
1077 }
1078
1079 if args$help {
1080   print_usage()
1081   exit(0)
1082 }
1083
1084 if args$version {
1085   print_version()
1086   exit(0)
1087 }
1088
1089 if args$help {
1090   print_usage()
1091   exit(0)
1092 }
1093
1094 if args$version {
1095   print_version()
1096   exit(0)
1097 }
1098
1099 if args$help {
1099 print_usage()
1100 exit(0)
1101 }
```

7|M-[M-^T 01, 25 10:17

## run-polap-r-bridge.R

Page 3/9

```

103 type = "integer",
104 default = 3000,
105 help = "Minimum length of pair-mapping alignment",
106 metavar = "number"
107 )
108 parser ← add_option(parser, c("-w", "--single-min"),
109 type = "integer",
110 default = 3000,
111 help = "Minimum length of single-mapping alignment",
112 metavar = "number"
113 )
114 parser ← add_option(parser, c("-x", "--bridge-min"),
115 type = "integer",
116 default = 3000,
117 help = "Minimum length of bridging reads",
118 metavar = "number"
119 )
120 parser ← add_option(parser, c("--intra-base-ratio"),
121 type = "double",
122 default = 0.7,
123 help = "Minimum of V10/V11",
124 metavar = "number"
125 )
126 parser ← add_option(parser, c("--intra-read-ratio"),
127 type = "double",
128 default = 0.7,
129 help = "Minimum of (V4-V3)/V2",
130 metavar = "number"
131 )
132 parser ← add_option(parser, c("--inter-base-ratio"),
133 type = "double",
134 default = 0.7,
135 help = "Minimum of V10/V11",
136 metavar = "number"
137 )
138 parser ← add_option(parser, c("--use-strand"),
139 action = "store_true",
140 default = FALSE, help = "will create ptgaul.names"
141 )
142 parser ← add_option(parser, c("--create-ptgaul"),
143 action = "store_true",
144 default = FALSE, help = "will create ptgaul.names"
145 )
146 parser ← add_option(parser, c("--create-single"),
147 action = "store_true",
148 default = FALSE, help = "will create single.names"
149 )
150 parser ← add_option(parser, c("--create-pair"),
151 action = "store_true",
152 default = FALSE, help = "will create pair.names"
153 )

```

7|M-[M-^T 01, 25 10:17

## run-polap-r-bridge.R

Page 4/9

```

154 parser ← add_option(parser, c("--create-combined"),
155   action = "store_true",
156   default = FALSE, help = "will create combined.names"
157 )
158 parser ← add_option(parser, c("--all"),
159   action = "store_true",
160   default = FALSE, help = "will create all the 4 names files."
161 )
162 parser ← add_option(parser, c("--outptgaul"),
163   action = "store",
164   default = "ptgaul.names",
165   help = "ptgaul output file"
166 )
167 parser ← add_option(parser, c("--outsingle"),
168   action = "store",
169   default = "single.names",
170   help = "single output file"
171 )
172 parser ← add_option(parser, c("--outpair"),
173   action = "store",
174   default = "pair.names",
175   help = "pair output file"
176 )
177 parser ← add_option(parser, c("--outcombined"),
178   action = "store",
179   default = "combined.names",
180   help = "combined output file"
181 )
182
183 args1 ← parse_args(parser)
184
185 if (is_null(args1$table)) {
186   s ← "bioprojects"
187   o ← "PRJNA817235-Canavalia_ensiformis"
188
189   input_dir0 ← file.path(".")
190   input1 ← file.path(input_dir0, "contig.tab")
191   input2 ← file.path(input_dir0, "mt.contig.name-1")
192   output1 ← file.path(input_dir0, "02-reads")
193
194   args1 ← parse_args(parser, args = c(
195     "--table", input1,
196     "--mtcontigname", input2,
197     "--o", output1,
198     "--r", 10000,
199     "--w", 10000,
200     "--x", 10000,
201     "--all",
202     "--use-strand",
203     "--intra-base-ratio", 0.7,
204     "--intra-read-ratio", 0.7,

```

7IM-^M-^T 01, 25 10:17

## run-polap-r-bridge.R

Page 5/9

```

205     "--inter-base-ratio", 0.7
206   })
207 }
208
209 # mt.contig.name-x
210 x1 <- as_tibble(read.table(args1$mtcontigname))
211
212 # https://lh3.github.io/minimap2/minimap2.html
213 # Li 2008 - "OUTPUT FORMAT: Minimap2 outputs mapping positions in the
214 # Pairwise mAPPING Format (PAF) by default. PAF is a TAB-delimited text format
215 # with each line consisting of at least 12 fields as are described in the
216 # following table:
217 #
218 # Col Type Description
219 # 1 string Query sequence name
220 # 2 int Query sequence length
221 # 3 int Query start coordinate (0-based)
222 # 4 int Query end coordinate (0-based)
223 # 5 char âM-^@M-^X+âM-^@M-^Y if query/target on the same strand; âM-^@M-^X-âM-^@M-^Y if opposite
224 # 6 string Target sequence name
225 # 7 int Target sequence length
226 # 8 int Target start coordinate on the original strand
227 # 9 int Target end coordinate on the original strand
228 # 10 int Number of matching bases in the mapping
229 # 11 int Number bases, including gaps, in the mapping
230 # 12 int Mapping quality (0-255 with 255 for missing)
231 #
232 # Read in the data with assigned column names
233 data <- read_tsv(args1$table,
234   show_col_types = FALSE,
235   col_names = c(
236     "rname",
237     "rlen",
238     "rstart",
239     "rend",
240     "strand",
241     "cname",
242     "clen",
243     "cstart",
244     "cend",
245     "match",
246     "base"
247   )
248 )
249
250 if (args1$`use-strand`) {
251   data <- data |>
252     filter(strand == "+")
253 }
254
255 mtdir <- args1$out

```

7|M-[M-^T 01, 25 10:17

**run-polap-r-bridge.R**

Page 6/9

```

256 pair_min ← as.numeric(args1$'pair-min')
257 brigde_min ← as.numeric(args1$'bridge-min')
258 single_min ← as.numeric(args1$'single-min')
259 a ← as.numeric(args1$'intra-read-ratio')
260 b ← as.numeric(args1$'intra-base-ratio')
261 c ← as.numeric(args1$'inter-base-ratio')
262 ptgaul_option_base ← paste("ptgaul",
263   a,
264   b,
265   single_min,
266   c,
267   pair_min,
268   brigde_min,
269   sep = "-"
270 )
271 ptgaul_option_base ← paste0(ptgaul_option_base, ".names")
272
273 single_option_base ← paste("single",
274   a,
275   b,
276   single_min,
277   c,
278   pair_min,
279   brigde_min,
280   sep = "-"
281 )
282 single_option_base ← paste0(single_option_base, ".names")
283
284 pair_option_base ← paste("pair",
285   a,
286   b,
287   single_min,
288   c,
289   pair_min,
290   brigde_min,
291   sep = "-"
292 )
293 pair_option_base ← paste0(pair_option_base, ".names")
294
295 combined_option_base ← paste("combined",
296   a,
297   b,
298   single_min,
299   c,
300   pair_min,
301   brigde_min,
302   sep = "-"
303 )
304 combined_option_base ← paste0(combined_option_base, ".names")
305
306 # 0. Check preconditions

```

7IM-^M-^T 01, 25 10:17

## run-polap-r-bridge.R

Page 7/9

```

307 # rstart is not greater than rend.
308 # match is not greater than base.
309 result <- data %>%
310   summarize(all_rows_meet_condition = all(rstart ≤ rend & match ≤ base))
311
312 # Check the result
313 stopifnot(result$all_rows_meet_condition)
314 stopifnot(pair_min ≥ 0, brigde_min ≥ 0, single_min ≥ 0)
315
316 # 1. ptGAUL: we use ptGAUL for the case of single edge reference
317 if (args1$'create-ptgaul' || args1$all) {
318   ptgaul_file <- file.path(mtdir, args1$outputptgaul)
319   ptgaul_mapped_reads <- data |>
320     filter(match / base > args1$'intra-base-ratio', base > single_min) |>
321     select(rname) |>
322     distinct(rname)
323
324   ptgaul_mapped_reads |>
325     write.table(
326       ptgaul_file,
327       row.names = FALSE,
328       col.names = FALSE,
329       quote = FALSE
330     )
331   ptgaul_option_file <- file.path(mtdir, ptgaul_option_base)
332   print(ptgaul_file)
333   file.copy(ptgaul_file, ptgaul_option_file)
334   # file.copy(ptgaul_file, ptgaul_option_file, showWarnings = FALSE)
335 }
336
337 # 2. intra-contig mapping
338 if (args1$'create-single' || args1$all) {
339   single_file <- file.path(mtdir, args1$outsingle)
340   intra_contig_mapped_reads <- data |>
341     filter(
342       (rend - rstart) / rlen > args1$'intra-read-ratio',
343       match / base > args1$'intra-base-ratio',
344       base > single_min
345     ) |>
346     select(rname) |>
347     distinct(rname)
348
349   intra_contig_mapped_reads |>
350     write.table(single_file, row.names = FALSE, col.names = FALSE, quote = FALSE)
351
352   single_option_file <- file.path(mtdir, single_option_base)
353   print(single_file)
354   file.copy(single_file, single_option_file)
355   # file.copy(single_file, single_option_file, showWarnings = FALSE)
356 }
357

```

7IM-^M-^T 01, 25 10:17

## run-polap-r-bridge.R

Page 8/9

```

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

```

7|M-[M-^T 01, 25 10:17

## run-polap-r-bridge.R

Page 9/9

```

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