

9IM-[M-^T 15, 25 17:43

**polap-bash-oatk-ont-sidekicks-stage2.sh**

Page 1/7

```

1  #!/usr/bin/env bash
2  # polap-bash-oatk-ont-sidekicks-stage12.sh
3  # Stage-1/2: use preprocessed reads from Stage-0, then:
4  #   Stage-1: (HPC default ON) âM-^FM-^R qc,assemble,summary  âM-^FM-^R backbone
5  #   Bait : backbone-based k-mer bait (bbduk|meryl)  âM-^FM-^R ONT mt-subset
6  #   Stage-2: recover (lift OR polish) on subset (chunked minimap2)  âM-^FM-^R annotate âM-^FM-^R pathfinder  âM-^FM-^R summary
7  #
8  # Inputs:
9  #   --reads-pre  OUT/pre/reads.pre.fq  (or .fa/.fq.gz)
10 #
11 # Requirements:
12 #   Always: seqkit, polap-bash-oatk-ont.sh
13 #   Bait: bbdruk.sh (default) OR meryl,meryl-lookup
14 #   Stage-2 mapping speed: handled by polap-bash-oatk-ont.sh via --reads-map / --map-chunks.
15
16 set -euo pipefail
17
18 : "${POLAP_LOG_LEVEL:=1}"
19 log() {
20     local l=$1
21     shift
22     [[ $POLAP_LOG_LEVEL -ge $l ]] && echo "$@" >&2
23 }
24 die() {
25     echo "[ERR] \"$@\" >&2
26     exit 1
27 }
28 need() { command -v "$1" >/dev/null 2>&1 || die "missing dependency: $1"; }
29
30 # ----- CLI defaults -----
31 READS_PRE=""
32 OUT=""
33 THREADS=32
34 OATKDB=""
35 CLADE="magnoliopsida"
36
37 # HPC for assembly (enabled by default)
38 HPC_ENABLE=1
39 HPC_OUT=""
40
41 # bait method
42 BAIT_METHOD="bbduk"
43 BBDUK_BIN="bbduk.sh"
44 MERYL_K=21
45 MAP_CHUNKS=4
46 MM2_EXTRA=""
47
48 # Step-wise runner
49 SCRIPT_DIR=$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
50 ONT_SCRIPT="${ONT_SCRIPT:-$SCRIPT_DIR/polap-bash-oatk-ont.sh}"
51

```

9IM-[M-^T 15, 25 17:43

**polap-bash-oatk-ont-sidekicks-stage2.sh**

Page 2/7

```

52 # Stage-1 assemble knobs
53 K_LIST_A="251,151,121,91"
54 SMER=31
55 AARC=0.25
56 WEAKX=0.20
57 UNZIP=6
58 NO_EC=0
59
60 # Stage-2 recovery
61 USE_LIFT=0
62 RACON_ROUNDS=2
63 MEDAKA_MODEL="r104_e81_sup_g615"
64 NO_MEDAKA=0
65
66 usage() {
67   cat <<EOF
68 Usage:
69   $(basename "$0") --reads-pre OUT/pre/reads.pre.fq --out OUTDIR --oatkdb /path/OatkDB [options]
70
71 Required:
72   --reads-pre FILE      preprocessed reads from Stage-0
73   --out     DIR          output folder (same root as Stage-0 is fine)
74   --oatkdb  DIR          OaktDB root (must have v20230921)
75
76 General:
77   --threads INT          (default ${THREADS})
78   --clade   NAME         fam prefix (default ${CLADE})
79   -v|--verbose           verbose
80   -q|--quiet             quiet
81
82 Assembly (Stage-1):
83   --hpc / --no-hpc       use seqtk hpc for assembly (default: on)
84   --hpc-out PATH         HPC FASTA path (default OUT/stage12/reads.hpc.fa)
85   --k-list "251,151,121,91"
86   --smert INT            (default ${SMER})
87   --a FLOAT              (default ${AARC})
88   --weak-cross FLOAT    (default ${WEAKX})
89   --unzip-round INT     (default ${UNZIP})
90   --no-ec                pass --no-read-ec
91
92 Bait:
93   --bait bbdruk|meryl   (default bbdruk)
94   --meryl-k INT          (default ${MERYL_K})
95
96 Recovery (Stage-2) + speed:
97   --lift                 use RLE lifter (Option B); else polish (Option A)
98   --racon-rounds INT    racon rounds (default ${RACON_ROUNDS})
99   --medaka-model STR    medaka model (default ${MEDAKA_MODEL})
100  --no-medaka           skip medaka
101  --map-chunks INT     chunk mapping (default ${MAP_CHUNKS})
102  --mm2-extra "FLAGS"   extra minimap2 flags to pass downstream

```

9IM-[M-^T 15, 25 17:43

**polap-bash-oatk-ont-sidekicks-stage2.sh**

Page 3/7

```
103 EOF
104 }
105
106 # ----- parse args -----
107 while [[ $# -gt 0 ]]; do
108     case "$1" in
109         --reads-pre)
110             READS_PRE="$2"
111             shift 2
112             ;;
113         --out)
114             OUT="$2"
115             shift 2
116             ;;
117         --oatkdb)
118             OATKDB="$2"
119             shift 2
120             ;;
121         --threads)
122             THREADS="$2"
123             shift 2
124             ;;
125         --clade)
126             CLADE="$2"
127             shift 2
128             ;;
129
130         --hpc)
131             HPC_ENABLE=1
132             shift
133             ;;
134         --no-hpc)
135             HPC_ENABLE=0
136             shift
137             ;;
138         --hpc-out)
139             HPC_OUT="$2"
140             shift 2
141             ;;
142
143         --bait)
144             BAIT_METHOD="$2"
145             shift 2
146             ;;
147         --meryl-k)
148             MERYL_K="$2"
149             shift 2
150             ;;
151
152         --k-list)
153             K_LIST_A="$2"
```

9|M-[M-^T 15, 25 17:43

**polap-bash-oatk-ont-sidekicks-stage2.sh**

Page 4/7

```

154     shift 2
155     ;;
156   --smcr)
157     SMER="$2"
158     shift 2
159     ;;
160   --a)
161     AARC="$2"
162     shift 2
163     ;;
164   --weak-cross)
165     WEAKX="$2"
166     shift 2
167     ;;
168   --unzip-round)
169     UNZIP="$2"
170     shift 2
171     ;;
172   --no-ec)
173     NO_EC=1
174     shift
175     ;;
176
177   --lift)
178     USE_LIFT=1
179     shift
180     ;;
181   --racon-rounds)
182     RACON_ROUNDS="$2"
183     shift 2
184     ;;
185   --medaka-model)
186     MEDAKA_MODEL="$2"
187     shift 2
188     ;;
189   --no-medaka)
190     NO_MEDAKA=1
191     shift
192     ;;
193
194   --map-chunks)
195     MAP_CHUNKS="$2"
196     shift 2
197     ;;
198   --mm2-extra)
199     MM2_EXTRA="$2"
200     shift 2
201     ;;
202
203   -v | --verbose)
204     POLAP_LOG_LEVEL=2

```

9IM-^M-^T 15, 25 17:43

**polap-bash-oatk-ont-sidekicks-stage2.sh**

Page 5/7

```

205     shift
206     ;;
207   -q | --quiet)
208     POLAP_LOG_LEVEL=0
209     shift
210     ;;
211   -h | --help)
212     usage
213     exit 0
214     ;;
215   *)
216     echo "[ERR] unknown arg: $1" >&2
217     usage
218     exit 1
219     ;;
220   esac
221 done
222 [[ -z "$READS_PRE" || -z "$OUT" || -z "$OATKDB" ]] && {
223   usage
224   exit 1
225 }
226
227 # ----- prep & deps -----
228 READS_PRE=$(readlink -f "$READS_PRE")
229 OUT=$(readlink -f "$OUT")
230 OATKDB=$(readlink -f "$OATKDB")
231 mkdir -p "$OUT/stage12"
232 cd "$OUT/stage12"
233
234 need seqkit
235 need "$ONT_SCRIPT"
236 [[ "$BAIT_METHOD" == "bbduk" ]] && need bbduk.sh || { [[ "$BAIT_METHOD" == "meryl" ]] && {
237   need meryl
238   need meryl-lookup
239 }; }
240 [[ $HPC_ENABLE -eq 1 ]] && need seqtk
241
242 # ----- Stage-1: assemble backbone -----
243 CUR="$READS_PRE"
244 HPC_FLAG=(--no-hpc)
245 CUR_ASM="$CUR"
246 if [[ $HPC_ENABLE -eq 1 ]]; then
247   HPC_FILE="${HPC_OUT}:$OUT/stage12/reads.hpc.fa"
248   log 1 "[hpc] seqtk hpc ^FM^R $HPC_FILE"
249   seqtk hpc "$CUR" >"$HPC_FILE"
250   CUR_ASM="$HPC_FILE"
251   HPC_FLAG=(--hpc --hpc-out "$HPC_FILE")
252 else
253   log 1 "[hpc] disabled for assembly"
254 fi
255

```

9iM-[M-^T 15, 25 17:43

**polap-bash-oatk-ont-sidekicks-stage2.sh**

Page 6/7

```

256 log 1 "[stage1] assemble backbone (k=${K_LIST_A}; smer=${SMER}; -a=${AARC})"
257 "$ONT_SCRIPT" \
258   --reads "$CUR_ASM" \
259   --out "$OUT/ont_stage1" \
260   --oatkdb "$OATKDB" --clade "$CLADE" \
261   --threads "$THREADS" \
262   "${HPC_FLAG[@]}" \
263   --k-list "$K_LIST_A" --smer "$SMER" --a "$AARC" --weak-cross "$WEAKX" --unzip-round "$UNZIP" \
264   ${[[ $NO_EC -eq 1 ]] && echo --no-read-ec} \
265   -s qc,assemble,summary -v
266
267 BACKBONE="$OUT/ont_stage1/k1/unitigs.fa"
268 [[ -s "$BACKBONE" ]] || die "backbone unitigs not found: $BACKBONE"
269
270 # ----- Bait: backbone-based ^M-^FM-^R subset -----
271 mkdir -p bait
272 SUBSET="$OUT/stage12/bait/ont.mt.fq"
273 mkdir -p "$(dirname "$SUBSET")"
274
275 if [[ "$BAIT_METHOD" == "bbduk" ]]; then
276   log 1 "[bait] bbduk: ref=$BACKBONE ^M-^FM-^R $SUBSET"
277   bbduk.sh in="$READS_PRE" outm="$SUBSET" outu="$OUT/stage12/bait/nonmt.fq" ref="$BACKBONE" k=31 hdist=1 threads="$THREADS"
278 else
279   log 1 "[bait] meryl k=$MERYL_K"
280   meryl k="$MERYL_K" count "$BACKBONE" output "$OUT/stage12/bait/seeds.k$MERYL_K.meryl"
281   meryl-lookup -existence -sequence "$READS_PRE" "$OUT/stage12/bait/seeds.k$MERYL_K.meryl" |
282     awk '/^>/ {print substr($0,2)}' >"$OUT/stage12/bait/mt.ids"
283   seqkit grep -f "$OUT/stage12/bait/mt.ids" "$READS_PRE" -o "$SUBSET"
284 fi
285 log 1 "[bait] subset size: $(wc -c <"$SUBSET") 2>/dev/null || echo 0) bytes"
286
287 # ----- Stage-2: recover (lift OR polish) -----
288 if [[ $USE_LIFT -eq 1 ]]; then
289   log 1 "[stage2] LIFT (RLE) + ^M-^W polish ^M-^FM-^R annotate ^M-^FM-^R PF"
290   "$ONT_SCRIPT" \
291     --reads "$CUR_ASM" \
292     --reads-map "$SUBSET" \
293     --map-chunks "$MAP_CHUNKS" --mm2-extra "$MM2_EXTRA" \
294     --out "$OUT/ont_stage2" \
295     --oatkdb "$OATKDB" --clade "$CLADE" \
296     --threads "$THREADS" \
297     --lift --polish-after-lift \
298     --racon-rounds 1 \
299   ${[[ $NO_MEDAKA -eq 1 ]] && echo --no-medaka || echo --medaka-model "$MEDAKA_MODEL"} \
300   -s lift,polish,annotate,pathfinder,summary -v
301 else
302   log 1 "[stage2] POLISH (racon/medaka) on subset ^M-^FM-^R annotate ^M-^FM-^R PF"
303   "$ONT_SCRIPT" \
304     --reads "$CUR_ASM" \
305     --reads-map "$SUBSET" \
306     --map-chunks "$MAP_CHUNKS" --mm2-extra "$MM2_EXTRA" \

```

9|M-[M-^T 15, 25 17:43

**polap-bash-oatk-ont-sidekicks-stage2.sh**

Page 7/7

```
307 --out "$OUT/ont_stage2" \
308 --oatkdb "$OATKDB" --clade "$CLADE" \
309 --threads "$THREADS" \
310 --racon-rounds "$RACON_ROUNDS" \
311 $([[ $NO_MEDAKA -eq 1 ]] && echo --no-medaka || echo --medaka-model "$MEDAKA_MODEL") \
312 -s polish,annotate,pathfinder,summary -v
313 fi
314
315 log 1 "[Stage-1/2 done]"
316 log 1 " backbone :$OUT/ont_stage1/k1/unitigs.fa"
317 log 1 " outputs :$OUT/ont_stage2 (polished/lifted, annotated, PF summary)"
```