```
#!/usr/bin/env bash
# polap-bash-oatk-ont-sidekicks-stage12.sh
# Stage-1/2: use preprocessed reads from Stage-0, then:
#   Stage-1: (HPC default ON) âM-^FM-^R qc,assemble,summary  âM-^FM-^R backbone
#   Bait   : backbone-based k-mer bait (bbduk|meryl) âM-^FM-^R ONT mt-subset
#   Stage-2: recover (lift OR polish) on subset (chunked minimap2) âM-^FM-^R annotate âM-^FM-^R pathfinder âM-^FM-^R summary
#
# Inputs:
#   --reads-pre  OUT/pre/reads.pre.fq  (or .fa/.fq.gz)
#
# Requirements:
#   Always: seqkit, polap-bash-oatk-ont.sh
#   Bait: bbduk.sh (default) OR meryl,meryl-lookup
#   Stage-2 mapping speed: handled by polap-bash-oatk-ont.sh via --reads-map / --map-chunks.

set -euo pipefail

: "${POLAP_LOG_LEVEL:=1}"
log() {
  local l=$1
  shift
  [[ $POLAP_LOG_LEVEL -ge $l ]] && echo "$@" >&2
}
die() {
  echo "[ERR]" "$@" >&2
  exit 1
}
need() { command -v "$1" >/dev/null 2>&1 || die "missing dependency: $1"; }

# -------------- CLI defaults --------------
READS_PRE=""
OUT=""
THREADS=32
OATKDB=""
CLADE="magnoliopsida"

# HPC for assembly (enabled by default)
HPC_ENABLE=1
HPC_OUT=""

# bait method
BAIT_METHOD="bbduk"
BBDUK_BIN="bbduk.sh"
MERYL_K=21
MAP_CHUNKS=4
MM2_EXTRA=""

# Step-wise runner
SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
ONT_SCRIPT="${ONT_SCRIPT:-"$SCRIPT_DIR/polap-bash-oatk-ont.sh"}"

```

```
# Stage-1 assemble knobs
K_LIST_A="251,151,121,91"
SMER=31
AARC=0.25
WEAKX=0.20
UNZIP=6
NO_EC=0

# Stage-2 recovery
USE_LIFT=0
RACON_ROUNDS=2
MEDAKA_MODEL="r104_e81_sup_g615"
NO_MEDAKA=0

usage() {
  cat <<EOF
Usage:
  $(basename "$0") --reads-pre OUT/pre/reads.pre.fq --out OUTDIR --oatkdb /path/OatkDB [options]

Required:
  --reads-pre FILE        preprocessed reads from Stage-0
  --out       DIR         output folder (same root as Stage-0 is fine)
  --oatkdb    DIR         OatkDB root (must have v20230921)

General:
  --threads INT           (default ${THREADS})
  --clade   NAME          fam prefix (default ${CLADE})
  -v|--verbose            verbose
  -q|--quiet              quiet

Assembly (Stage-1):
  --hpc / --no-hpc        use seqtk hpc for assembly (default: on)
  --hpc-out PATH          HPC FASTA path (default OUT/stage12/reads.hpc.fa)
  --k-list "251,151,121,91"
  --smer INT              (default ${SMER})
  --a FLOAT               (default ${AARC})
  --weak-cross FLOAT      (default ${WEAKX})
  --unzip-round INT       (default ${UNZIP})
  --no-ec                 pass --no-read-ec

Bait:
  --bait bbduk|meryl      (default bbduk)
  --meryl-k INT           (default ${MERYL_K})

Recovery (Stage-2) + speed:
  --lift                  use RLE lifter (Option B); else polish (Option A)
  --racon-rounds INT      racon rounds (default ${RACON_ROUNDS})
  --medaka-model STR      medaka model (default ${MEDAKA_MODEL})
  --no-medaka             skip medaka
  --map-chunks INT        chunk mapping (default ${MAP_CHUNKS})
  --mm2-extra "FLAGS"     extra minimap2 flags to pass downstream
```

```
EOF
}

# -------------- parse args --------------
while [[ $# -gt 0 ]]; do
  case "$1" in
  --reads-pre)
    READS_PRE="$2"
    shift 2
    ;;
  --out)
    OUT="$2"
    shift 2
    ;;
  --oatkdb)
    OATKDB="$2"
    shift 2
    ;;
  --threads)
    THREADS="$2"
    shift 2
    ;;
  --clade)
    CLADE="$2"
    shift 2
    ;;

  --hpc)
    HPC_ENABLE=1
    shift
    ;;
  --no-hpc)
    HPC_ENABLE=0
    shift
    ;;
  --hpc-out)
    HPC_OUT="$2"
    shift 2
    ;;

  --bait)
    BAIT_METHOD="$2"
    shift 2
    ;;
  --meryl-k)
    MERYL_K="$2"
    shift 2
    ;;

  --k-list)
    K_LIST_A="$2"
```

```
    shift 2
    ;;
  --smer)
    SMER="$2"
    shift 2
    ;;
  --a)
    AARC="$2"
    shift 2
    ;;
  --weak-cross)
    WEAKX="$2"
    shift 2
    ;;
  --unzip-round)
    UNZIP="$2"
    shift 2
    ;;
  --no-ec)
    NO_EC=1
    shift
    ;;

  --lift)
    USE_LIFT=1
    shift
    ;;
  --racon-rounds)
    RACON_ROUNDS="$2"
    shift 2
    ;;
  --medaka-model)
    MEDAKA_MODEL="$2"
    shift 2
    ;;
  --no-medaka)
    NO_MEDAKA=1
    shift
    ;;

  --map-chunks)
    MAP_CHUNKS="$2"
    shift 2
    ;;
  --mm2-extra)
    MM2_EXTRA="$2"
    shift 2
    ;;

  -v | --verbose)
    POLAP_LOG_LEVEL=2
```

```
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 âM−^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
```

```
log 1 "[stage1] assemble backbone (k=${K_LIST_A}; smer=${SMER}; −a=${AARC})"
"$ONT_SCRIPT" \
  --reads "$CUR_ASM" \
  --out "$OUT/ont_stage1" \
  --oatkdb "$OATKDB" --clade "$CLADE" \
  --threads "$THREADS" \
  "${HPC_FLAG[@]}" \
  --k-list "$K_LIST_A" --smer "$SMER" --a "$AARC" --weak-cross "$WEAKX" --unzip-round "$UNZIP" \
  $([[ $NO_EC -eq 1 ]] && echo --no-read-ec) \
  -s qc,assemble,summary -v

BACKBONE="$OUT/ont_stage1/k1/unitigs.fa"
[[ -s "$BACKBONE" ]] || die "backbone unitigs not found: $BACKBONE"

# -------------- Bait: backbone-based âM-^FM-^R subset --------------
mkdir -p bait
SUBSET="$OUT/stage12/bait/ont.mt.fq"
mkdir -p "$(dirname "$SUBSET")"

if [[ "$BAIT_METHOD" == "bbduk" ]]; then
  log 1 "[bait] bbduk: ref=$BACKBONE âM−^FM−^R $SUBSET"
  bbduk.sh in="$READS_PRE" outm="$SUBSET" outu="$OUT/stage12/bait/nonmt.fq" ref="$BACKBONE" k=31 hdist=1 threads="$THREADS"
else
  log 1 "[bait] meryl k=$MERYL_K"
  meryl k="$MERYL_K" count "$BACKBONE" output "$OUT/stage12/bait/seeds.k$MERYL_K.meryl"
  meryl-lookup -existence -sequence "$READS_PRE" "$OUT/stage12/bait/seeds.k$MERYL_K.meryl" |
    awk '/^>/{print substr($0,2)}' >"$OUT/stage12/bait/mt.ids"
  seqkit grep -f "$OUT/stage12/bait/mt.ids" "$READS_PRE" -o "$SUBSET"
fi
log 1 "[bait] subset size: $( (wc -c <"$SUBSET") 2>/dev/null || echo 0) bytes"

# -------------- Stage-2: recover (lift OR polish) --------------
if [[ $USE_LIFT -eq 1 ]]; then
  log 1 "[stage2] LIFT (RLE) + 1ÃM−^W polish âM−^FM−^R annotate âM−^FM−^R PF"
  "$ONT_SCRIPT" \
    --reads "$CUR_ASM" \
    --reads-map "$SUBSET" \
    --map-chunks "$MAP_CHUNKS" --mm2-extra "$MM2_EXTRA" \
    --out "$OUT/ont_stage2" \
    --oatkdb "$OATKDB" --clade "$CLADE" \
    --threads "$THREADS" \
    --lift --polish-after-lift \
    --racon-rounds 1 \
    $([[ $NO_MEDAKA -eq 1 ]] && echo --no-medaka || echo --medaka-model "$MEDAKA_MODEL") \
    -s lift,polish,annotate,pathfinder,summary -v
else
  log 1 "[stage2] POLISH (racon/medaka) on subset âM−^FM−^R annotate âM−^FM−^R PF"
  "$ONT_SCRIPT" \
    --reads "$CUR_ASM" \
    --reads-map "$SUBSET" \
    --map-chunks "$MAP_CHUNKS" --mm2-extra "$MM2_EXTRA" \
```

```
    --out "$OUT/ont_stage2" \
    --oatkdb "$OATKDB" --clade "$CLADE" \
    --threads "$THREADS" \
    --racon-rounds "$RACON_ROUNDS" \
    $([[ $NO_MEDAKA -eq 1 ]] && echo --no-medaka || echo --medaka-model "$MEDAKA_MODEL") \
    -s polish,annotate,pathfinder,summary -v
fi

log 1 "[Stage-1/2 done]"
log 1 "  backbone : $OUT/ont_stage1/k1/unitigs.fa"
log 1 "  outputs  : $OUT/ont_stage2 (polished/lifted, annotated, PF summary)"
```