```
#!/usr/bin/env bash
# polap-bash-fast-mtseed-ont.sh v0.3.0
# Version: v0.3.0 (merged PT-removal + fast mtseed; steps start at 1)
#
# Pipeline outline (ONT):
#   1) Remove plastid-origin reads (PT) using isoform A/B + identity cutoff
#      (auto-learned or --identity-min override) âM−^FM−^R reads.nonpt.fq.gz
#      [optional: --profiling]
#   2) All-vs-all overlap:
#        âM−^@¢ default (classic) OR
#        âM−^@¢ --overlap-mode scan+stream+edges  (light scan âM−^FM−^R shortlist âM−^FM−^R streamed edges)
#      (also: run miniprot vs BUSCO on a 10% subsample for QC only, if −n is given)
#   3) Select mito-enriched reads (nuclear-aware if --nuc-ids given) from overlapness
#   4) Assemble selected reads (miniasm | raven)
#   5) Intergenic filtering (miniprot vs mito CDS; read-recruit median gate)
#   6) Map readsâM−^FM−^Rseeds (if iterating)
#   7) Stop when ÎM−^T|M| < threshold (if iterating)
#
# Required helpers (must exist; not embedded here):
#   ${_POLAPLIB_DIR}/polap-bash-pt-isoform.sh
#   ${_POLAPLIB_DIR}/remove-ptdna-reads/polap-py-pt-ident-threshold.py
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-py-overlapness-from-paf.py
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-py-topfrac-by-wdeg.py
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-py-filter-paf-by-ids.py
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-py-cds-coverage-from-paf.py
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-py-seed-mapped-reads.py
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-py-stop-delta.py
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-awk-filter-conservative.awk
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-bash-recruit-count.sh
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-bash-paf2mreads.sh
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-r-pick-high-recruit.R
#   ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-py-threshold-from-nuclear.py
#   (for scan+stream+edges mode)
#     ${_POLAPLIB_DIR}/fast-mtseed-ont/paf-topk-per-q.py
#     ${_POLAPLIB_DIR}/fast-mtseed-ont/paf-to-edges-stream.py
#     ${_POLAPLIB_DIR}/fast-mtseed-ont/polap-py-edges-components.py  (optional)
#
set −euo pipefail

# âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
# Paths & environment
# âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
_POLAPLIB_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
```

```
44  source "${_POLAPLIB_DIR}/polap−lib−conda.sh" || true
45  HELP="${_POLAPLIB_DIR}/fast−mtseed−ont"
46  mkdir −p "${HELP}"
47
48  # Helpers (fast−mtseed)
49  OLPY="${HELP}/polap−py−overlapness−from−paf.py"
50  TOPPY="${HELP}/polap−py−topfrac−by−wdeg.py"
51  FILTPY="${HELP}/polap−py−filter−paf−by−ids.py"
52  CDSPY="${HELP}/polap−py−cds−coverage−from−paf.py"
53  SEEDREADSPY="${HELP}/polap−py−seed−mapped−reads.py"
54  STOPPY="${HELP}/polap−py−stop−delta.py"
55  AWK_CONS="${HELP}/polap−awk−filter−conservative.awk"
56  RECRUIT_SH="${HELP}/polap−bash−recruit−count.sh"
57  PAF2MREADS_SH="${HELP}/polap−bash−paf2mreads.sh"
58  R_PICK="${HELP}/polap−r−pick−high−recruit.R"
59  THRESH_NUC_PY="${HELP}/polap−py−threshold−from−nuclear.py"
60  PY_PT_THRESH="${HELP}/polap−py−pt−ident−threshold.py"
61
62  # Helpers for PT removal & isoforms
63  PT_ISOFORM_SH="${_POLAPLIB_DIR}/polap−bash−pt−isoform.sh"
64
65  # scan+stream+edges mode
66  TOPKPY="${HELP}/paf−topk−per−q.py"
67  EDGPY="${HELP}/paf−to−edges−stream.py"
68  EDGECOMP="${HELP}/polap−py−edges−components.py" # optional
69
70  # âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
71  # Logging & profiling
72  # âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
73  VERBOSE=1
74  DRY=0
75  LOG_FD="/dev/stderr"
76  TEE_ACTIVE=0
77  PROF=0
78  PROF_FILE=""
79
80  # Base helper for all note levels
81  _note_base() {
82    local lvl=$1 # message verbosity level
83    shift
84    local msg="$*"
85
86    # Format with timestamp + caller
```

```
87    local src="${BASH_SOURCE[2]##*/}" # 2 to skip _note_base + noteX
88    local ln="${BASH_LINENO[1]}"
89    local ts
90    ts="$(date +'%F %T')"
91    local line="[$ts][${src}:${ln}] $msg"
92
93    # Always log, but only display if verbosity allows
94    if [[ "${VERBOSE:−0}" −gt "$lvl" ]]; then
95      if [[ "${_arg_log_stderr:−off}" == "off" ]]; then
96        printf "%s\n" "$line" >&3 # real screen
97      else
98        printf "%s\n" "$line" >&2
99      fi
100   fi
101
102   # Always append to log (stdout goes to logit, or LOG if you like)
103   printf "%s\n" "$line"
104 }
105
106 # Note functions
107 note() { _note_base 1 "$@"; }
108 note0() { _note_base 0 "$@"; } # always
109 note1() { _note_base 1 "$@"; } # shown with −v
110 note2() { _note_base 2 "$@"; } # shown with −v −v
111 note3() { _note_base 3 "$@"; } # shown with −v −v −v
112
113 cmd() {
114   if ((DRY)); then
115     local p="$1"
116     shift
117     printf "[DRY] %q" "$p" | tee −a "$LOG_FD"
118     for a in "$@"; do printf " %q" "$a" | tee −a "$LOG_FD"; done
119     printf "\n" | tee −a "$LOG_FD"
120     return 0
121   fi
122   local p="$1"
123   shift
124   printf "[RUN] %q" "$p" | tee −a "$LOG_FD"
125   for a in "$@"; do printf " %q" "$a" | tee −a "$LOG_FD"; done
126   printf "\n" | tee −a "$LOG_FD"
127   "$p" "$@"
128 }
129 prof_start() { ((PROF)) && PROF_T0=$(date +%s) && PROF_STEP="$1" || true; }
130 prof_end() {
131   ((PROF)) || return 0
132   local t1=$(date +%s)
133   echo −e "${PROF_STEP}\t$((t1 − PROF_T0))" >>"$PROF_FILE"
134 }
135
136 require_tools() {
137   local miss=0
```

```
  for t in "$@"; do
    command -v "$t" >/dev/null 2>&1 || {
      note "ERR missing tool: $t"
      miss=1
    }
  done
  if ((miss == 1)); then
    note "ERR load modules/conda"
    exit 127
  fi
}

# âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
# CLI (merged options)
# âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
reads=""
pt_ref=""
outdir="fast_mtseed_ont_out"
threads=32
# PT removal knobs
pt_origin=""
nuc_origin=""
alen_min=3000
identity_min="" # override identity if set
fpr=0.01
tpr=0.95
profiling=0
# All-vs-all / selection / assembly
# overlap_mode="default" # default | scan+stream+edges
overlap_mode="scan+stream+edges"
scan_keep_frac="0.40"
scan_k=21
scan_w=9
scan_N=1
scan_mask="0.70"
scan_minocc=20
refine_k=19
refine_w=7
refine_N=30
refine_mask="0.60"
refine_minocc=10
topk_per_q=0
topk_edges_per_node=0
```

```
keep_scan_paf=0
gzip_edges=1
top_frac="0.20"
min_olen=1200
min_ident="0.84"
w_floor="0.12"
assembler="miniasm"
# intergenic / BUSCO
mcds=""
nprot=""
nuc_ids_opt=""
# iteration
rounds_max=1
delta_stop="0.05"
seed=13
steps=""
# nuclear overlap deplete is handled in separate tool; here we select seeds only

usage() {
  cat <<EOF
polap-bash-fast-mtseed-ont.sh v0.3.0 (merged PT removal + fast mtseed)
Usage:
  $0 -r reads.fq.gz -p pt_ref.fa -o outdir [options]

Required:
  -r, --reads                 ONT reads (FASTQ[.gz])
  -p, --pt-ref                plastid reference FASTA (single circle)

PT removal:
  --pt-origin FILE           PT-origin read IDs (weak labels) [optional]
  --mt-origin FILE          Nuclear-origin read IDs (weak labels) [optional]
  --alen-min INT             aligned length guard [${alen_min}]
  --identity-min FLOAT       override auto identity cutoff (e.g., 0.93)
  --fpr FLOAT                nuclear FPR target for identity [${fpr}]
  --tpr FLOAT                PT TPR target for identity [${tpr}]

All-vs-all:
  --overlap-mode MODE        default | scan+stream+edges   [${overlap_mode}]
  --scan-keep-frac FLOAT     shortlist fraction by wdegree [${scan_keep_frac}]
  --scan-k INT  --scan-w INT --scan-N INT --scan-mask F --scan-minocc INT
  --refine-k INT --refine-w INT --refine-N INT --refine-mask F --refine-minocc INT
  --topk-per-q INT           cap scan hits per query (0=off)
  --topk-edges-per-node INT  cap edges per node in reducer (0=off)
  --keep-scan-paf            store scan.paf.gz for QA
  --no-gzip-edges            keep edges.tsv uncompressed
  --top-frac FLOAT           top fraction if no nuclear IDs [${top_frac}]
  --min-olen INT --min-ident F --w-floor F   edge filters [${min_olen},${min_ident},${w_floor}]

Intergenic & assembly:
  -m, --mt-cds FILE          mito CDS proteins (miniprot) [optional]
  -n, --nuc-prot FILE        BUSCO proteins (unused here; leave for future)
```

```
232   --assembler NAME           miniasm|raven [${assembler}]
233   --rounds-max INT           iteration rounds [${rounds_max}]
234   --delta-stop FLOAT         stop if ÎM−^T|M| < x [${delta_stop}]
235
236 General:
237   -o, --outdir DIR           output directory [${outdir}]
238   -t, --threads INT          threads [${threads}]
239   --seed INT                 RNG seed [${seed}]
240   -v, --verbose              increase verbosity
241   --quiet                    quiet
242   --dry                      dry-run (no execution)
243   --profiling                per-step timings (log/profile.tsv)
244   -h, --help                 help
245 EOF
246 }
247
248 while [[ $# -gt 0 ]]; do
249   case "$1" in
250   -r | --reads)
251     reads="$2"
252     shift 2
253     ;;
254   --step)
255     steps="$2"
256     shift 2
257     ;;
258   -p | --pt-ref)
259     pt_ref="$2"
260     shift 2
261     ;;
262   -o | --outdir)
263     outdir="$2"
264     shift 2
265     ;;
266   --nuc-ids)
267     nuc_ids_opt="$2"
268     shift 2
269     ;;
270   -t | --threads)
271     threads="$2"
272     shift 2
273     ;;
274   --pt-origin)
275     pt_origin="$2"
276     shift 2
277     ;;
278   --mt-origin)
279     mt_origin="$2"
280     shift 2
281     ;;
282   --nuc-origin)
```

```
283       nuc_origin="$2"
284       shift 2
285       ;;
286     --alen-min)
287       alen_min="$2"
288       shift 2
289       ;;
290     --identity-min)
291       identity_min="$2"
292       shift 2
293       ;;
294     --fpr)
295       fpr="$2"
296       shift 2
297       ;;
298     --tpr)
299       tpr="$2"
300       shift 2
301       ;;
302     --overlap-mode)
303       overlap_mode="$2"
304       shift 2
305       ;;
306     --scan-keep-frac)
307       scan_keep_frac="$2"
308       shift 2
309       ;;
310     --scan-k)
311       scan_k="$2"
312       shift 2
313       ;;
314     --scan-w)
315       scan_w="$2"
316       shift 2
317       ;;
318     --scan-N)
319       scan_N="$2"
320       shift 2
321       ;;
322     --scan-mask)
323       scan_mask="$2"
324       shift 2
325       ;;
326     --scan-minocc)
327       scan_minocc="$2"
328       shift 2
329       ;;
330     --refine-k)
331       refine_k="$2"
332       shift 2
333       ;;
```

```
  --refine-w)
    refine_w="$2"
    shift 2
    ;;
  --refine-N)
    refine_N="$2"
    shift 2
    ;;
  --refine-mask)
    refine_mask="$2"
    shift 2
    ;;
  --refine-minocc)
    refine_minocc="$2"
    shift 2
    ;;
  --topk-per-q)
    topk_per_q="$2"
    shift 2
    ;;
  --topk-edges-per-node)
    topk_edges_per_node="$2"
    shift 2
    ;;
  --keep-scan-paf)
    keep_scan_paf=1
    shift
    ;;
  --no-gzip-edges)
    gzip_edges=0
    shift
    ;;
  --top-frac)
    top_frac="$2"
    shift 2
    ;;
  --min-olen)
    min_olen="$2"
    shift 2
    ;;
  --min-ident)
    min_ident="$2"
    shift 2
    ;;
  --w-floor)
    w_floor="$2"
    shift 2
    ;;
  -m | --mt-cds)
    mcds="$2"
    shift 2
```

```
    ;;
  -n | --nuc-prot)
    nprot="$2"
    shift 2
    ;;
  --rounds-max)
    rounds_max="$2"
    shift 2
    ;;
  --delta-stop)
    delta_stop="$2"
    shift 2
    ;;
  --seed)
    seed="$2"
    shift 2
    ;;
  -v | --verbose)
    VERBOSE=$((VERBOSE + 1))
    shift
    ;;
  --quiet | -q)
    VERBOSE=0
    shift
    ;;
  --dry)
    DRY=1
    shift
    ;;
  --profiling)
    PROF=1
    shift
    ;;
  -h | --help)
    usage
    exit 0
    ;;
  *)
    note "ERR unknown arg: $1"
    usage
    exit 2
    ;;
  esac
done

require_tools minimap2 samtools seqkit python Rscript awk sort comm cut gzip

for s in "$OLPY" "$TOPPY" "$FILTPY" "$CDSPY" "$SEEDREADSPY" \
  "$STOPPY" "$AWK_CONS" "$RECRUIT_SH" "$PAF2MREADS_SH" \
  "$R_PICK" "$THRESH_NUC_PY" "$PT_ISOFORM_SH" "$PY_PT_THRESH"; do
  [[ -s "$s" ]] || {
```

```
436       note "ERR missing helper: $s"
437       exit 1
438     }
439   done
440   if [[ "$overlap_mode" == "scan+stream+edges" ]]; then
441     for s in "$TOPKPY" "$EDGPY"; do [[ -s "$s" ]] || {
442       note "ERR missing $s"
443       exit 1
444     }; done
445   fi
446
447   [[ -n "$reads" && -s "$reads" ]] || {
448     note "ERR missing −−reads"
449     exit 2
450   }
451   [[ -n "$pt_ref" && -s "$pt_ref" ]] || {
452     note "ERR missing −−pt−ref"
453     exit 2
454   }
455
456   mkdir -p "$outdir" "$outdir/log"
457   LOG="${outdir}/pipeline.log"
458   LOG_FD="$LOG"
459   if ((!DRY)); then
460     exec > >(tee -a "$LOG") 2>&1
461     TEE_ACTIVE=1
462   fi
463   if ((PROF)); then
464     PROF_FILE="$outdir/log/profile.tsv"
465     echo -e "step\tseconds" >"$PROF_FILE"
466   fi
467
468   note "Outdir: $outdir"
469   note "Assembler: $assembler ; threads: $threads"
470   note "Overlap mode: $overlap_mode"
471
472   # Step gating
473   _step_set=""
474   _add_steps() {
475     local spec="$1"
476     IFS=',' read -r -a arr <<<"$spec"
477     for tok in "${arr[@]}"; do
478       if [[ "$tok" =~ ^([0-9]+)-([0-9]+)$ ]]; then
479         local a=${BASH_REMATCH[1]} b=${BASH_REMATCH[2]}
480         for ((i = a; i <= b; i++)); do _step_set+=" $i "; done
481       elif [[ "$tok" =~ ^[0-9]+$ ]]; then _step_set+=" $tok "; fi
482     done
483   }
484   _should_run() {
485     local s="$1"
486     [[ -z "$steps" ]] && return 0
```

```
487   [[ −z "$_step_set" ]] && _add_steps "$steps"
488   [[ " $_step_set " == *" $s "* ]]
489 }
490
491 if _should_run 0; then
492   note "note"
493   note0 "note0"
494   note1 "note1"
495   note2 "note2"
496   note3 "note3"
497 fi
498
499 # âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
500 # Step 1: PT removal (organized)
501 # âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
502 R1="$reads"
503 if _should_run 1; then
504   # 1a) Build PT isoforms A/B (handles IR/SSC/LSC & origin)
505   note0 "1a) Build PT isoforms A/B (handles IR/SSC/LSC & origin)"
506   PANEL_DIR="$outdir/panel"
507   mkdir −p "$PANEL_DIR"
508   prof_start "pt_isoform"
509   cmd bash "$PT_ISOFORM_SH" −r "$pt_ref" −o "$PANEL_DIR" −t "$threads"
510   prof_end
511   ISO_A="$PANEL_DIR/pt_isomerA.fa"
512   ISO_B="$PANEL_DIR/pt_isomerB.fa"
513   [[ −s "$ISO_A" ]] || {
514     note "ERR missing pt_isomerA.fa"
515     exit 3
516   }
517
518   note1 "A: $ISO_A"
519   [[ −s "$ISO_B" ]] && note1 "B: $ISO_B"
520
521   # 1b) Make single−record doubled references (A/B)
522   note0 "1b) Make single−record doubled references (A/B)"
523   dbld() {
524     python − "$1" "$2" <<'PY'
525 import sys
526 inp,outp=sys.argv[1],sys.argv[2]
527 name=None; seq=[]
528 with open(inp) as f:
529   for ln in f:
```

```
    if ln.startswith('>'):
      if name is None: name=ln[1:].strip().split()[0]
      else: break
    else: seq.append(ln.strip())
S=''.join(seq)
with open(outp,'w') as w:
  w.write('>pt.double\n'); w.write(S+S+'\n')
PY
  }
  DBL_A="$PANEL_DIR/pt_isomerA.double.fa"
  DBL_B="$PANEL_DIR/pt_isomerB.double.fa"
  prof_start "pt_double"
  ((!DRY)) && dbld "$ISO_A" "$DBL_A"
  ((!DRY)) && { [[ -s "$ISO_B" ]] && dbld "$ISO_B" "$DBL_B" || true; }
  prof_end
  note1 "A: $DBL_A"
  [[ -s "$DBL_B" ]] && note1 "B: $DBL_B"

  # 1c) Map readsâM−^FM−^Rdoubled A/B
  note0 "1c) Map readsâM−^FM−^R doubled A/B"
  PAF_A="$outdir/map/formA.paf"
  PAF_B="$outdir/map/formB.paf"
  mkdir -p "$outdir/map"
  prof_start "map_ptA"
  note "Step1: map readsâM−^FM−^R isomerA.double > $PAF_A"
  cmd minimap2 -x map-ont --secondary=yes -N 50 -t "$threads" "$DBL_A" "$reads" >"$PAF_A"
  prof_end
  if [[ -s "$DBL_B" ]]; then
    prof_start "map_ptB"
    note "Step1: map readsâM−^FM−^R isomerB.double > $PAF_B"
    cmd minimap2 -x map-ont --secondary=yes -N 50 -t "$threads" "$DBL_B" "$reads" >"$PAF_B"
    prof_end
  fi

  # 1d) Determine identity cutoff (MT-guided) & emit pt.ids
  note0 "1d) Determine identity cutoff (MT−guided) & emit pt.ids"
  PT_VARS="$outdir/pt_thresh.vars"
  PT_DIAG="$outdir/pt_thresh.diag.tsv"
  PT_IDS="$outdir/pt.ids"
  if [[ -n "$identity_min" ]]; then
    note "Step1: using −−identity−min=$identity_min ; alen_min=$alen_min"
    prof_start "emit_pt_ids_override"
    if ((!DRY)); then
      awk -v ID="$identity_min" -v AL="$alen_min" 'BEGIN{FS=OFS="\t"}
    NF>=12{ id=($11>0?$10/$11:0); if(id>=ID && $11+0>=AL) print $1 }' \
        "$PAF_A" | sort -u >"$outdir/ptA.ids"
      if [[ -s "$PAF_B" ]]; then
        awk -v ID="$identity_min" -v AL="$alen_min" 'BEGIN{FS=OFS="\t"}
     NF>=12{ id=($11>0?$10/$11:0); if(id>=ID && $11+0>=AL) print $1 }' \
          "$PAF_B" | sort -u >"$outdir/ptB.ids"
        sort -u "$outdir/ptA.ids" "$outdir/ptB.ids" >"$PT_IDS"
```

```
581         else
582           mv "$outdir/ptA.ids" "$PT_IDS"
583         fi
584         printf "ident_min=%s\nalen_min=%d\n" "$identity_min" "$alen_min" >"$PT_VARS"
585         : >"$PT_DIAG"
586       fi
587       prof_end
588     else
589       prof_start "pt_ident_threshold"
590       if ((!DRY)); then
591         python "$PY_PT_THRESH" \
592           --paf "$PAF_A" ${PAF_B:+ "$PAF_B"} \
593           --pt-ids "${pt_origin:-/dev/null}" \
594           --mt-ids "${mt_origin:-/dev/null}" \
595           --alen-min "$alen_min" \
596           --fpr "$fpr" --tpr "$tpr" \
597           --diag "$PT_DIAG" \
598           --emit-pt-ids "$PT_IDS" >"$PT_VARS"
599         note "PT IDs: $PT_IDS"
600         note "PT Vars: $PT_VARS"
601       fi
602       prof_end
603     fi
604
605     # 1e) Remove PT reads âM−^FM−^R R1
606     note0 "1e) Remove PT reads âM−^FM−^R R1"
607     ident_min_out="$identity_min"
608     if [[ -s "$PT_VARS" ]]; then
609       while IFS== read -r kv; do
610         case "$kv" in
611         ident_min=*) ident_min_out="${kv#ident_min=}" ;;
612         alen_min=*) alen_min="${kv#alen_min=}" ;;
613         esac
614       done <"$PT_VARS"
615     fi
616     note "Step1: PT thresholds âM−^FM−^R ident_min=${ident_min_out} alen_min=${alen_min}"
617     prof_start "write_nonPT"
618     if ((!DRY)); then
619       seqkit fx2tab -ni "$reads" | sort -u >"$outdir/map/all.ids"
620       sort -u "$PT_IDS" >"$outdir/map/pt.ids.sorted"
621       comm -23 "$outdir/map/all.ids" "$outdir/map/pt.ids.sorted" >"$outdir/keep.nonpt.ids"
622       note "Step1: write $outdir/reads.nonpt.fq.gz"
623       seqkit grep -f "$outdir/keep.nonpt.ids" "$reads" -o "$outdir/reads.nonpt.fq.gz"
624     fi
625     prof_end
626     R1="$outdir/reads.nonpt.fq.gz"
627   fi
628   [[ -s "$R1" ]] || {
629     note "ERR missing R1 ($R1)"
630     exit 3
631   }
```

```
632
633 # âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
634 # Step 2: all−vs−all (organized) + optional BUSCO on 10% sample (QC)
635 # âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
    M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
    M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
636 ST1="$outdir/01_allvsall"
637 mkdir −p "$ST1"
638 PAFALL="$ST1/allvsall.paf"
639 OTSV_GLOBAL="$ST1/overlapness.tsv"
640 EDGES_GLOBAL="$ST1/edges.tsv.gz"
641
642 if _should_run 2; then
643   if [[ "$overlap_mode" != "scan+stream+edges" ]]; then
644     note0 "Step2: minimap2 ava−ont R1ÃM−^WR1 > $PAFALL"
645     ((!DRY)) && minimap2 −x ava−ont −t "$threads" --secondary=yes −N 50 \
646       --mask−level 0.50 "$R1" "$R1" >"$PAFALL"
647   else
648     note0 "Step2: overlap−mode scan+stream+edges"
649     SCAN_DIR="$ST1/scan"
650     REF_DIR="$ST1/refine"
651     mkdir −p "$SCAN_DIR" "$REF_DIR"
652
653     # 2a) scan (light) âM−^FM−^R shortlist
654     note0 "2a) scan (light) âM−^FM−^R  shortlist"
655     prof_start "scan"
656     if ((!DRY)); then
657       if ((keep_scan_paf)); then
658         minimap2 −t "$threads" −x ava−ont \
659           −k"$scan_k" −w"$scan_w" −N "$scan_N" \
660           --mask−level "$scan_mask" --min−occ−floor "$scan_minocc" \
661           "$R1" "$R1" |
662           { ((topk_per_q > 0)) && python "$TOPKPY" "$topk_per_q" || cat; } |
663           tee >(gzip −1 >"$SCAN_DIR/scan.paf.gz") |
664           python "$OLPY" − --min_olen 800 --min_ident 0.82 --w_floor 0.08 \
665             >"$SCAN_DIR/scan.overlapness.tsv"
666       else
667         minimap2 −t "$threads" −x ava−ont \
668           −k"$scan_k" −w"$scan_w" −N "$scan_N" \
669           --mask−level "$scan_mask" --min−occ−floor "$scan_minocc" \
670           "$R1" "$R1" |
671           { ((topk_per_q > 0)) && python "$TOPKPY" "$topk_per_q" || cat; } |
672           python "$OLPY" − --min_olen 800 --min_ident 0.82 --w_floor 0.08 \
673             >"$SCAN_DIR/scan.overlapness.tsv"
674       fi
```

```
675         awk 'NR>1{print $1"\t"$3}' "$SCAN_DIR/scan.overlapness.tsv" |
676           sort -k2,2nr |
677           awk -v f="$scan_keep_frac" '{a[NR]=$1} END{lim=int(NR*f); for(i=1;i<=lim;i++) print a[i]}' \
678             >"$SCAN_DIR/shortlist.ids"
679         seqkit grep -f "$SCAN_DIR/shortlist.ids" "$R1" -o "$SCAN_DIR/R1.short.fq.gz"
680       fi
681       prof_end
682
683       # 2b) refine streamed âM-^FM-^R  edges + overlapness
684       note0 "2b) refine streamed âM-^FM-^R  edges + overlapness"
685       prof_start "refine"
686       if ((!DRY)); then
687         minimap2 -t "$threads" -x ava-ont \
688           -k"$refine_k" -w"$refine_w" --secondary=yes -N "$refine_N" \
689           --mask-level "$refine_mask" --min-occ-floor "$refine_minocc" \
690           "$SCAN_DIR/R1.short.fq.gz" "$SCAN_DIR/R1.short.fq.gz" |
691           python "$EDGPY" \
692             --min-olen "$min_olen" --min-ident "$min_ident" --w-floor "$w_floor" \
693             --topk-node "$topk_edges_per_node" \
694             --edges "$ST1/edges.tsv" --overlapness "$OTSV_GLOBAL"
695         ((gzip_edges)) && gzip -f "$ST1/edges.tsv"
696       fi
697       prof_end
698       EDGES_GLOBAL="$ST1/edges.tsv${gzip_edges:+.gz}"
699
700       # optional graph summary
701       if ((!DRY)) && [[ -s "$EDGECOMP" ]]; then
702         COMP_DIR="$ST1/comp"
703         mkdir -p "$COMP_DIR"
704         python "$EDGECOMP" "$EDGES_GLOBAL" \
705           --summary "$COMP_DIR/summary.tsv" \
706           --sizes "$COMP_DIR/comp_sizes.tsv" \
707           --membership "$COMP_DIR/node2comp.tsv" \
708           --deg "$COMP_DIR/degree.tsv" \
709           --giant-nodes "$COMP_DIR/giant_nodes.txt" \
710           --giant-edges "$COMP_DIR/giant_edges.tsv" \
711           --print || true
712       fi
713     fi
714
715     # 2c) BUSCO on a 10% length-stratified subsample (QC only)
716     note0 "2c) BUSCO on a 10% length−stratified subsample (QC only)"
717     if [[ -n "$nprot" ]]; then
718       note "Step2: miniprot BUSCO on 10% subsample of PT−depleted reads (QC)"
719       SAMP_IDS="$ST1/nonpt.sample.ids"
720       SAMP_FQ="$ST1/reads.nonpt.sample.fq.gz"
721       # simple random 10% by seed (keeps it fast and reproducible)
722       ((!DRY)) && seqkit sample -s "$seed" -p 0.10 "$R1" -o "$SAMP_FQ"
723       ((!DRY)) && miniprot -d "$ST1/nonpt.sample.mpi" "$SAMP_FQ"
724       ((!DRY)) && miniprot -t "$threads" -S -N 3 --outc 0.4 \
725         "$ST1/nonpt.sample.mpi" "$nprot" >"$ST1/nonpt.sample.busco.paf"
```

```
726       ((!DRY)) && awk 'BEGIN{FS=OFS="\t"} NF>=12 && $11+0>=150 {print $1}' \
727         "$ST1/nonpt.sample.busco.paf" | sort -u >"$ST1/nuc.ids.sample"
728     fi
729   fi
730
731   # âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
      M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
      M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
      M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
      M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
732   # Step 3: overlapness + selection (organized)
733   # âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
      M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
      M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
      M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
      M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
734   RDIR="$outdir/round_1"
735   mkdir -p "$RDIR"
736   if _should_run 3; then
737
738     OTSV="$RDIR/overlapness.tsv"
739     note0 "3a) Prepare $OTSV"
740     if [[ "$overlap_mode" == "scan+stream+edges" ]]; then
741       note "Step3: reuse refined overlapness âM−^FM−^R  $OTSV"
742       ((!DRY)) && cp -f "$OTSV_GLOBAL" "$OTSV"
743     else
744       note "Step3: compute overlapness from PAF âM−^FM−^R  $OTSV"
745       ((!DRY)) && python "$OLPY" "$PAFALL" \
746         --min_olen "$min_olen" --min_ident "$min_ident" --w_floor "$w_floor" \
747         >"$OTSV"
748     fi
749
750     # --- Step 3: selection (prefer BUSCO sample labels from Step 2c) ---
751     note0 "3b) selection (prefer BUSCO sample labels from Step 2c)"
752     SELECT_IDS="$RDIR/select_ids.txt"
753
754     # Prefer the miniprot BUSCO sample labels from Step 2c, else fallback to --nuc-ids
755     NUC_SAMPLE="$ST1/nuc.ids.sample"
756     NUC_FOR_THRESH=""
757     if [[ -s "$NUC_SAMPLE" ]]; then
758       NUC_FOR_THRESH="$NUC_SAMPLE"
759     elif [[ -n "$nuc_ids_opt" && -s "$nuc_ids_opt" ]]; then
760       NUC_FOR_THRESH="$nuc_ids_opt"
761     fi
762
763     if [[ -n "$NUC_FOR_THRESH" ]]; then
764       note "Step3: nuclear-guided selection using $NUC_FOR_THRESH -> $SELECT_IDS"
765       if ((!DRY)); then
766         python "$THRESH_NUC_PY" "$OTSV" "$NUC_FOR_THRESH" \
767           --mode fpr --fpr 0.01 --diag "$RDIR/threshold_from_nuclear.tsv" \
768           >"$RDIR/nuc_thresh.vars"
```

```
769
770         wdeg_min=0
771         deg_min=0
772         while IFS== read -r kv; do
773           case "$kv" in
774           wdeg_min=*) wdeg_min="${kv#wdeg_min=}" ;;
775           deg_min=*) deg_min="${kv#deg_min=}" ;;
776           esac
777         done <"$RDIR/nuc_thresh.vars"
778
779         awk -v W="$wdeg_min" -v D="$deg_min" 'BEGIN{FS=OFS="\t"}
780     NR>1 && ($3+0)>=W && ($2+0)>=D {print $1}' "$OTSV" |
781         sort -u >"$RDIR/organelle.ids"
782
783         # Exclude labeled nuclear reads (from sample or full set)
784         comm -23 <(sort -u "$RDIR/organelle.ids") \
785           <(sort -u "$NUC_FOR_THRESH") \
786           >"$SELECT_IDS"
787       fi
788     else
789       note "Step3: select top-frac ($top_frac) by wdegree"
790       ((!DRY)) && python "$TOPPY" "$OTSV" --top_frac "$top_frac" >"$SELECT_IDS"
791     fi
792
793   fi
794
795   # âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@â
       M-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^T
       M-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@â
       M-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^T
       M-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@
   # Step 4: assemble selected reads (organized)
   # âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@â
       M-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^T
       M-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@â
       M-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^T
       M-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@âM-^TM-^@
   SEEDS_ROUND=""
   if _should_run 4; then
     if [[ "$assembler" == "miniasm" ]]; then
       SUBPAF="$RDIR/top_overlaps.paf.gz"
       # In streamed mode we do not have a refined PAF; miniasm will still run with -f
       if [[ "$overlap_mode" != "scan+stream+edges" ]]; then
         note "Step4: filter PAF by selected ids âM-^FM-^R $SUBPAF"
         ((!DRY)) && python "$FILTPY" "$PAFALL" "$RDIR/select_ids.txt" -o "$SUBPAF"
       fi
       TOPFA="$RDIR/top_reads.fa.gz"
       note "Step4: extract selected reads âM-^FM-^R $TOPFA"
       ((!DRY)) && seqkit fq2fa "$R1" | seqkit grep -f "$RDIR/select_ids.txt" -o "$TOPFA"
       GFA="$RDIR/miniasm.gfa"
       note "Step4: run miniasm on the selected reads"
```

```
812       ((!DRY)) && miniasm -f "$TOPFA" <(zcat -f "$SUBPAF" 2>/dev/null || true) >"$GFA" || true
813       SEEDS_ROUND="$RDIR/m_seeds_raw.fa"
814       note "Step4: extract contigs from GFA âM−^FM−^R $SEEDS_ROUND"
815       ((!DRY)) && awk '/^S/{print ">"$2"\n"$3}' "$GFA" >"$SEEDS_ROUND" || :
816     else
817       TOPFQ="$RDIR/top_reads.fq.gz"
818       note "Step4: extract selected reads (fq) âM−^FM−^R $TOPFQ"
819       ((!DRY)) && seqkit grep -f "$RDIR/select_ids.txt" "$R1" -o "$TOPFQ"
820       SEEDS_ROUND="$RDIR/raven_contigs.fasta"
821       note "Step4: raven −−disable−polishing âM−^FM−^R $SEEDS_ROUND"
822       ((!DRY)) && raven --threads "$threads" --disable-polishing -o "$SEEDS_ROUND" "$TOPFQ"
823     fi
824     # ensure downstream finalize has a source
825     SEEDS_CUR="$SEEDS_ROUND"
826   fi
827
828   # âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
      M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
      M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
      M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
      M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
   # Step 4: assemble selected reads
   #   Input  : select_ids.txt (IDs of mito−enriched reads, from Step 3)
   #   Output : contigs assembled from those reads
   #
   #   Miniasm requires:
   #     (1) the read sequences (FASTA/FASTQ)
   #     (2) the overlaps between those reads (PAF)
   #
   #   Raven can do both overlap + assembly internally, so it only needs reads.
   # âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
      M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
      M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@â
      M−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^T
      M−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@âM−^TM−^@
   SEEDS_ROUND=""
   if _should_run 4; then
     SELECT_IDS="$RDIR/select_ids.txt"

     if [[ "$assembler" == "miniasm" ]]; then
       # Extract the actual sequences of the selected reads
       TOPFA="$RDIR/top_reads.fa.gz"
       note "Step4: extract selected reads âM−^FM−^R $TOPFA"
       ((!DRY)) && seqkit fq2fa "$R1" | seqkit grep -f "$SELECT_IDS" -o "$TOPFA"

       GFA="$RDIR/miniasm.gfa"

       if [[ "$overlap_mode" != "scan+stream+edges" ]]; then
         # DEFAULT MODE:
         # In Step 2 we produced a full all-vs-all PAF (PAFALL).
         # Here we filter it down to overlaps *only between the selected reads*.
```

```
855         SUBPAF="$RDIR/top_overlaps.paf.gz"
856         note "Step4: filter full PAF to selected read overlaps âM−^FM−^R $SUBPAF"
857         ((!DRY)) && python "$FILTPY" "$PAFALL" "$SELECT_IDS" -o "$SUBPAF"
858
859         note "Step4: run miniasm with filtered overlaps"
860         ((!DRY)) && miniasm -f "$TOPFA" <(zcat -f "$SUBPAF") >"$GFA"
861
862       else
863         # STREAMED MODE:
864         # In Step 2 we *did not* save a full PAF (we streamed directly to reducers).
865         # Miniasm still needs overlaps, so we regenerate them âM−^@M−^T but only among
866         # the smaller set of selected reads.
867         SELPAF="$RDIR/selected_allvsall.paf.gz"
868         note "Step4: generate overlaps among selected reads âM−^FM−^R $SELPAF"
869         ((!DRY)) && minimap2 -x ava-ont -t "$threads" --secondary=yes -N 30 \
870           --mask-level 0.60 --min-occ-floor 10 \
871           "$TOPFA" "$TOPFA" | gzip -1 >"$SELPAF"
872
873         note "Step4: run miniasm with selected−read overlaps"
874         ((!DRY)) && miniasm -f "$TOPFA" <(zcat -f "$SELPAF") >"$GFA"
875       fi
876
877       # Extract contigs from the GFA into a FASTA for downstream use
878       SEEDS_ROUND="$RDIR/m_seeds_raw.fa"
879       note "Step4: extract contigs from GFA âM−^FM−^R $SEEDS_ROUND"
880       ((!DRY)) && awk '/^S/{print ">"$2"\n"$3}' "$GFA" >"$SEEDS_ROUND" || :
881
882     else
883       # RAVEN MODE:
884       # Raven overlaps + assembles internally. We only need to provide the reads.
885       TOPFQ="$RDIR/top_reads.fq.gz"
886       note "Step4: extract selected reads (fq) âM−^FM−^R $TOPFQ"
887       ((!DRY)) && seqkit grep -f "$SELECT_IDS" "$R1" -o "$TOPFQ"
888
889       SEEDS_ROUND="$RDIR/raven_contigs.fasta"
890       note "Step4: raven −−disable−polishing âM−^FM−^R $SEEDS_ROUND"
891       ((!DRY)) && raven --threads "$threads" --disable-polishing -o "$SEEDS_ROUND" "$TOPFQ"
892     fi
893
894     # Save path for finalize step
895     SEEDS_CUR="$SEEDS_ROUND"
896   fi
897
898   if _should_run 5; then
899     note "polap readassemble"
900
901     # annotate miniasm assembly contigs
902     #
903     contigger_dir="${RDIR}/30−contigger"
904     mkdir -p "${contigger_dir}"
905     sed 's/LN:i/dp:i/' "${RDIR}/miniasm.gfa" >"${contigger_dir}/graph_final.gfa"
```

```
906
907     # polap annotate -o "${outdir}" -i round_1
908
909     # select seed contigs
910     #
911     bash "${_POLAPLIB_DIR}/../polap2.sh" readassemble annotated \
912       -o "${outdir}" -i round_1 \
913       -l "${reads}"
914
915   fi
916
917   # Finalize
918   if ((!DRY)); then
919     FINAL_DIR="$(dirname "$SEEDS_CUR")"
920     FINAL_SEEDS="$FINAL_DIR/m_seeds_final.fa"
921     note "Finalize: cp $SEEDS_CUR âM−^FM−^R $FINAL_SEEDS"
922     cp "$SEEDS_CUR" "$FINAL_SEEDS"
923     note "Done. Final seeds: $FINAL_SEEDS"
924   else
925     note "−−dry finished (no commands executed)."
926   fi
```