mosdepth批量质控：计算靶区覆盖及20X碱基比例

用mosdepth处理多样本BAM，固定BED和过滤口径，汇总按碱基加权的20X覆盖比例，并以样本清单和日志验收。

批量覆盖质控要先统一统计口径。全基因组平均深度、每个目标区间平均深度和达到20X的目标碱基比例回答不同问题。本文以坐标排序的人类WES BAM为例，使用一份已核对、排序且互不重叠的target BED；通过mosdepth输出区域和阈值结果，再按长度加权汇总。若BED区间重叠，先按项目规则处理，否则重叠碱基会重复计入分母。

项目	做法	注意
样本清单	唯一ID与BAM绝对路径	避免同名目录覆盖
区域	固定排序、不重叠target BED	核对参考与坐标
指标	均值、区域阈值数、加权比例	不要相互替代
线程	mosdepth解压线程	不等于整机并发额度

按相同口径运行样本

samples.tsv无表头，两列TAB分隔。先填写TARGETS和THREADS环境变量，在一个工作目录运行。下面使用10、20、30倍阈值并关闭逐碱基输出；MAPQ及排除标记沿用项目固定规则，若需修改应记录并建立新批次。需处理CRAM时按官方说明提供匹配参考。

```
set -euo pipefail
: "${TARGETS:?■■■■■■■target BED}"
: "${THREADS:?■■■■■■■■}"
case "$THREADS" in ""|*[!0-9]*) exit 2;; esac
test "$THREADS" -ge 1
test -s "$TARGETS" && test -s samples.tsv
mkdir depth-batch-001
root="$PWD/depth-batch-001"
mosdepth --version > "$root/version.txt" 2>&1
completed=0
while IFS=$'■' read -r sample bam || [[ -n "$sample$bam" ]]; do
  case "$sample" in ""|*[!A-Za-z0-9_-]*) exit 2;; esac
  test -s "$bam"; mkdir "$root/$sample"
  samtools quickcheck "$bam"
  sha256sum "$bam" "$TARGETS" > "$root/$sample/inputs.sha256"
  /usr/bin/time -v -o "$root/$sample/resources.txt"     mosdepth --threads "$THREADS" --by
"$TARGETS" --thresholds 10,20,30     --no-per-base "$root/$sample/depth" "$bam"     >
"$root/$sample/run.log" 2>&1
  for suffix in mosdepth.summary.txt regions.bed.gz thresholds.bed.gz; do test -s "$root/
$sample/depth.$suffix"; done
  completed=$((completed+1))
done < samples.tsv
test "$completed" -gt 0
```

用区间长度计算整体达标比例

mosdepth.summary.txt给出染色体和全局汇总，regions.bed.gz对应各目标区域的平均深度，thresholds.bed.gz记录每个目标达到阈值的碱基数。汇总全项目比例应使用最后一项，并以相同的target总长度作为分母。其他阈值可另行计算并分别标注，保存计算记录便于查验。

阈值文件每行提供目标区间在指定深度以上的碱基数。下面汇总器要求目标有序且不重叠，并计算20X碱基总数除以靶区总长度；它不是“平均深度超过20X的区间比例”。例如两个区间长度分别为10和30 bp，达到20X的碱基数分别为5和15，则合计比例是20/40=50%；直接平均两个区间比例会得到75%，口径就错了。所有样本必须已有完整输出，结果文件以独占方式创建，避免覆盖旧批次。

```
import csv, gzip
from pathlib import Path
with open("samples.tsv", newline="") as f:
    plan = list(csv.reader(f, delimiter="\t"))
assert plan and all(len(row) == 2 for row in plan)
assert len({row[0] for row in plan}) == len(plan)
results = []
for sample, _ in plan:
    path = Path("depth-batch-001") / sample / "depth.thresholds.bed.gz"
    total = covered = 0; ends = {}
    with gzip.open(path, "rt") as f:
        rows = csv.reader(f, delimiter="\t"); header = next(rows); ix = header.index("20X")
        for row in rows:
            chrom, start, end, count = row[0], int(row[1]), int(row[2]), int(row[ix])
            assert end > start >= 0 and 0 <= count <= end-start
            assert start >= ends.get(chrom, 0), "overlapping/unsorted targets"
            ends[chrom] = end; total += end-start; covered += count
    assert total > 0
    results.append([sample, total, covered, covered/total])
with open("depth-qc.csv", "x", newline="") as f:
    w = csv.writer(f); w.writerow(["sample","target_bases","bases_ge20","fraction_ge20"]);
w.writerows(results)
```

解释异常并安排并发

先确认summary、regions和thresholds文件都来自本次运行；quickcheck只检查BAM基本结构，不验证样本身份、参考是否匹配或科学结果合理。若某样本20X比例低，检查测序量、比对、捕获区和过滤口径；若全批一起偏低，先复查BED、坐标、参考及参数。均值高不能证明每个目标都覆盖充分，建议保留区域结果定位低覆盖缺口。需要比较样本时确认它们使用同一组目标、同一阈值和同一过滤设置；目标有变更就分组汇报，不把缺失区间当成零覆盖悄悄合并。

记录样本ID、输入散列、BED散列、mosdepth版本、线程、过滤参数、运行时间、峰值资源及结果状态。先做串行基线再测少量并发；线程主要控制BAM解压，实际加速还受I/O和样本大小影响。失败保留日志并用新目录重跑，同时区分BAM读取、磁盘不足、线程配置和目标文件问题；汇总器拒绝重复ID、重叠区域及缺失输出，避免旧结果混入。报告比例时同时提供达标碱基数和靶区总长度，方便复核分子、分母及批次差异。

参考资料

mosdepth官方README：选项及输出文件 官方文档