NextPolish批量校正：reads映射、轮次管理和结果核验

完整操作教程 · 参考固定 · 往返恢复验收

NextPolish批量校正不能把同一份BAM反复交给不同轮次。本篇固定NextPolish 1.4.1的短读长路线：以一份初始组装和一对已质控的双端reads，先完成可追溯试跑，再扩到多个组装。NextPolish2及其他reads路线另有接口，不混用配置。

运行边界：示例代码需按安装版本和真实项目输入复核；未提供真实生物数据，语法或小样本检查不代表科学结果和性能验收。

NextPolish批量校正不能把同一份BAM反复交给不同轮次。本篇固定NextPolish 1.4.1的短读长路线：以一份初始组装和一对已质控的双端reads，先完成可追溯试跑，再扩到多个组装。NextPolish2及其他reads路线另有接口，不混用配置。

先区分校正步骤与失败重试

安装环境须已通过官方测试数据，并具备该版本需要的BWA、samtools等依赖。配置里的task安排校正模块，rerun控制失败任务重试，两者不是同一概念。两个模块按序执行时，后续映射应以相应阶段的组装为参考；直接复用旧参考的比对会破坏这个前提。

项目	本例设置	含义
task	1212	两组1→2短读长模块，按官方任务定义执行
rerun	0	先暴露失败原因，非校正轮数
parallel_jobs / multithread_jobs	1 / 2	教学试跑值，非通用配置
rewrite / deltmp	no / no	保留失败现场与中间证据

为一个组装建立独立工作区

以下在Linux Bash执行。三个输入使用无空白的绝对路径；reads来自同一配对文库，已检查接头、污染和配对关系。复制组装用于保护原始版本，reads保持只读引用。sgs.fofn每行一个文件，R1、R2按对相邻排列，不写成一行两个路径。

```
set -euo pipefail
r1=/data/A_R1.fastq.gz
r2=/data/A_R2.fastq.gz
asm=/data/assembly-A.fa
for f in "$r1" "$r2" "$asm"; do
  test -s "$f"
  [[ "$f" != *[[:space:]]* ]]
done
gzip -t "$r1" "$r2"
mkdir polish-A-001
root="$PWD/polish-A-001"
[[ "$root" != *[[:space:]]* ]]
cp "$asm" "$root/input.fa"
printf '%s\n' "$r1" "$r2" > "$root/sgs.fofn"
cat > "$root/run.cfg" <<EOF
[General]
job_type = local
job_prefix = polish_A
task = 1212
rewrite = no
deltmp = no
rerun = 0
parallel_jobs = 1
multithread_jobs = 2
genome = $root/input.fa
genome_size = auto
workdir = $root/work
polish_options = -p {multithread_jobs}
[sgs_option]
sgs_fofn = $root/sgs.fofn
```

```
sgs_options = -max_depth 100 -bwa
EOF
sha256sum "$root/input.fa" "$r1" "$r2" "$root/run.cfg" > "$root/inputs.sha256"
/usr/bin/time -v -o "$root/run.time" nextPolish "$root/run.cfg" \
  > "$root/console.txt" 2>&1
test -s "$root/work/genome.nextpolish.fasta"
test -s "$root/work/genome.nextpolish.fasta.stat"
```

先检查完整轮次，再评价序列

读取console和子任务日志，确认配置中的模块均结束，输出确由本次输入产生；仅有某个BAM或FASTA不能证明整条任务完成。官方最终输出位于workdir，统计文件记录校正信息。保留软件版本及安装来源，可用原始输入的SHA重新确认本次任务没有读错组装。

不要仅比较N50判断校正好坏。比较校正前后总长度、序列名称、校正位置与reads支持，结合独立的完整性或错误评估。输出中的大小写可能携带处理信息，不为了格式统一直接覆盖最终文件；确需转换时另存派生文件并记录操作。

失败和批量扩展的处理顺序

失败先查首个未完成模块的命令、参考索引及磁盘，再查reads路径和依赖版本；不要通过增加rerun掩盖稳定复现的输入错误。更换reads、参数或初始组装时建立新目录，避免rewrite设置与旧中间结果造成混合。每个组装保留自己的配置、FOFN、日志和结果，不能让多个任务共写一个workdir。

先记录一个完整任务的墙钟、RSS及工作盘高水位。外层同时启动多个组装会叠加内部子任务，不能只用multithread_jobs估计整机占用。GNU time的RSS不能完整代表多进程同时峰值，必要时从调度器或cgroup读取任务总峰值。完成单组装后，每次只增加一个并发层级，并确认队列中每项都有最终FASTA、统计和检查记录。

gzip -t只验证压缩内容可解码，不检查R1/R2记录数和名称对应。提交前另外核对配对记录数、首末记录及文库来源，FOFN增加多个文库时也按对登记。验收表逐项写原始组装SHA、文库、task字符串、完成模块、最终FASTA及统计路径；下一轮显式以哪一份序列为参考，也要能从日志回查。

配套练习包

本PDF对应的ZIP包含本文代码块、主题练习模板或合成数据、README和SHA256校验文件。请从同一教程页面下载练习包；先核对校验值，再在独立目录复现。

参考资料

NextPolish任务配置 · 来源资料

NextPolish官方教程 · 来源资料

NextPolish 1.4.1源码与配置 · 来源资料

操作建议：先在合成小样本上验证流程，再依据项目参考、标签合同、存储策略与获配资源验收真实数据。