核对工具的流式输入支持并测试压缩数据处理链

完整操作教程 · 参考固定 · 往返恢复验收

用严格四行FASTQ验证器串接gzip和seqtk，避免落盘完整解压文件，并保留管道失败状态。

运行边界：仅在下游明确接受FASTA且不再需要质量值时使用转换步骤。

能否边解压边分析，取决于目标工具是否接受标准输入，以及它是否需要回读、索引或多次扫描。本篇选择一个清楚的小任务：将单端、严格四行格式的FASTQ.gz转换为压缩FASTA。只适用于下游确实只要序列的情况；FASTA会丢弃质量值，不能用它替代需要质量信息的分析输入。

先写清管道两端的合同

需要GNU gzip、Bash pipefail、Python3和支持标准输入的seqtk。原始数据先校验源端文件指纹。seqtk原生支持gzip文件，但这里显式解压以展示管道的错误传播；不生成完整未压缩FASTQ。双端读长还需要配对ID及同步检查，本例不把两份文件随意拼成一条流。

步骤	接收或产生	验收内容
gzip -dc	压缩文件到字节流	CRC/截断错误返回非零
四行验证器	逐条读记录，仅输出计数	头部、加号、非空序列和质量长度
seqtk seq -a -	标准输入到FASTA	确定下游允许丢弃质量
gzip -c	FASTA流到压缩工作文件	管道整体成功后才改最终名

保存一个常量内存的结构检查器

将下段保存为check4.py。它逐记录读取，不把全部reads放入内存；遇到多行FASTQ会拒绝，因此先确认数据确实符合四行合同。它只检查基本结构，不等于碱基质量、污染、接头或研究范围验收。

```
import sys
n=0
f=sys.stdin
while True:
    head=f.readline()
    if not head:break
    seq=f.readline().rstrip('\r\n')
    plus=f.readline();qual=f.readline().rstrip('\r\n')
    assert head.startswith('@') and head[1:].strip()
    assert plus.startswith('+') and seq and len(seq)==len(qual)
    n+=1
assert n>0
print(n)
```

验证与转换各读一遍，不落盘解压副本

```
#!/usr/bin/env bash
set -euo pipefail
input=/data/S1.fastq.gz
test -s "$input"
test -s check4.py
mkdir streaming-001
root="$PWD/streaming-001"
gzip -dc "$input" | python3 check4.py > "$root/input-records.txt"
gzip -dc "$input" | seqtk seq -a - | gzip -c > "$root/sequences.fa.gz.partial"
gzip -t "$root/sequences.fa.gz.partial"
gzip -dc "$root/sequences.fa.gz.partial" | \
  awk '/^>/{n++} END{print n+0}' > "$root/output-records.txt"
cmp "$root/input-records.txt" "$root/output-records.txt"
mv "$root/sequences.fa.gz.partial" "$root/sequences.fa.gz"
```

只有输入验证、转换管道、压缩完整性和记录数一致全部通过，才出现最终文件名。Bash默认只返回管道最后一个进程状态，pipefail能让上游失败影响整体结果；仅看gzip写出了文件容易漏掉前端错误。失败时保留.partial与错误日志，不能把它交给下游当完整输出。

这两遍读取用额外I/O换取先检查再转换，适合不愿保存解压副本的流程，并不保证最快。正式运行同时记录压缩输入、压缩输出、工具内部临时文件和后续索引的空间；标准输入只省掉一个解压文件，不会自动消除所有中间产物。

若目标程序只接受普通路径、需要随机访问或要求成对多轮读取，应使用它正式支持的压缩接口、缓存或分批方案。命名管道也不是随机访问文件，不能把路径参数能打开FIFO当成整个算法都支持。先用已知小样例比较流式与常规路线的序列及结果，再判断是否仍需工作盘。

确认FASTQ合同，不要把流式等同于安全转换

FASTQ常见的是四行一条记录，但输入也可能是多行序列/质量格式。四行检查器遇到合法多行数据会拒绝，因而运行前需依据数据来源和格式说明确认合同。还应确认文件完整、gzip流能读到末尾、记录数非零、每条序列与质量长度相等、头行标识非空。双端数据需按读段ID核对配对关系，不能只比较R1/R2记录总数。

FASTQ转FASTA会删除质量值。若下游软件需要质量评分、质量过滤、错误模型或读取质量分布，就不应转换成FASTA。转换前写下目标程序的输入格式要求、下游是否允许丢失质量、是否需要配对信息和是否要求随机访问。若只是避免额外解压副本，很多工具能直接读gzip输入；应先查工具对应版本是否原生支持压缩输入，避免为了“流式”而多做一次转换。

用最小样例验收管道语义

建立一个极小FASTQ夹具，包括多条记录、相同序列和不同长度，并人为制作截断压缩文件及错误质量长度样例。正常样例应得到预期记录数；损坏样例必须非零退出，且不能留下正式输出文件。逐条转换后核对每个FASTA序列与输入序列相同，同时明确FASTA输出头部是否保留read ID以及注释部分如何处理。只比记录数无法发现序列内容被截断或ID被意外改写。

管道使用临时后缀写结果，全部步骤成功后再原子改名。Bash的pipefail使前序命令失败能传播到整条管道；仍要保存退出状态和stderr。输出写到与最终结果同一文件系统，原子改名才不会跨盘复制。运行目录须唯一，避免覆盖之前的有效结果；失败时保留日志和partial用于排查，但下游清单只接受正式文件名。

估算峰值空间与计算成本

流式减少的是完整未压缩FASTQ副本，不表示工作空间为零。临时FASTA.gz、输入源、输出索引、日志和下游中间文件可能同时存在。若压缩文件大小为S，先在代表输入上测量转换输出大小、墙钟、CPU和磁盘读写，再按批次并发估算峰值；不要假设FASTA必然比FASTQ小，因为质量字段被移除但序列熵与压缩行为也影响体积。若输出需保留，正式容量至少需覆盖输入、结果和备份窗口。

一次完整验证再转换会读两遍源文件，用额外I/O换取提前发现格式问题；对于慢速对象存储或校内共享盘，这可能明显延长时间。可以在同一条管道中做校验和转换，但必须保证验证器失败能中止写出、临时文件不会被误当完成，并对失败样例进行覆盖测试。选择哪条路线应由数据大小、I/O成本、工具能力和失败恢复要求决定，而不是只追求命令更短。

故障定位和恢复执行

症状	检查证据	安全处理
gzip CRC或截断报错	源端SHA、完整下载状态	隔离输入并重新取得完整对象
四行检查失败	首条异常记录和格式合同	确认是否多行FASTQ或质量长度问题
seqtk非零退出	版本、参数与stderr	用小样例验证正确输入形式
输出记录数不同	管道状态、过滤行为和文件结尾	保留partial，不发布最终文件
写盘失败	容量、inode、配额和挂载点	停止新任务并确认输出目录

重新运行时换新尝试目录或确认只清理本次产生的partial，不覆盖已验收文件。两端FASTQ成对时，R1、R2必须作为一组共同校验和提交；其中一个失败，不应把另一个标为可供分析。数据通过结构检查仍需按研究目标运行质控、污染检查和下游分析，这些科学判断不由格式转换脚本完成。

何时需要专用计算工作区

如果常规计算节点可以安全读取压缩数据，转换不需要额外独占服务器；若多组数据并行、需要稳定工作盘、长时间运行或共享目录经常耗尽容量，再评估按项目使用隔离工作环境。沟通时给出FASTQ数量与总字节、是否双端、目标格式、并行批次、保留时间、当前文件系统和代表任务实测，不需上传原始样本。资源建议取决于真实I/O和临时空间，而不是只按测序数据类型直接推荐高配机器。

双端数据要作为不可拆分的一组

对paired-end样本，批次清单明确记录R1、R2路径、各自字节数和哈希，并写明命名解析规则。读段ID中的配对标记格式会因来源而不同，必须抽查真实标题并核对工具规则；不能简单删除末尾字符后假定匹配。将转换后的FASTA继续用于下游前，应确认分析是否需要质量值、是否要求成对输入以及FASTA头部如何识别read mate。若工具只接受FASTQ，不应因为转换减少文件或省空间就强行改变格式。

验收小样本时逐对比较输入read ID和输出ID，同时比较序列字符串和记录数。对大型文件可用流式脚本完成比对并输出首个差异位置，不把所有读段载入内存。若序列顺序由转换工具保持，则逐条比较最直接；若工具会排序或过滤，需先定义允许的变化和稳定匹配键，并把过滤规则纳入交付记录。通过计数但序列错位时，仍然是失败。

控制临时文件与失败恢复

将每次尝试放在唯一目录，分别保存校验计数、stderr、命令行、版本、输入指纹、转换输出临时文件和最终状态。正式文件名只在全流程验证通过后出现。若失败原因是文件系统满，先查看目标盘和inode，确认partial由本次运行生成后再处理；不要删除用户目录里所有同后缀文件。清单应把状态写为failed或verified，重跑后保留旧attempt记录。

如果管道连接多个工具，分别确认每个工具支持从标准输入读取、是否会自行缓存整份数据、是否需要随机访问、是否隐式过滤记录。一个程序接受stdin并不代表整个下游链都是常量内存。用进程监控和实际峰值记录验证，而不是从命令外观推断“流式”。必要时先运行短样例并观察临时目录变化。

根据下游目的选择路线

路线A是保留原FASTQ.gz并直接交给支持gzip输入的程序；它避免转换副本并保留质量。路线B是先结构检查再转FASTA，适用于下游明确只需要序列、且质量信息不再使用的情形。路线C是按样本或区间分片处理，适用于工具支持分块而整体输入/输出过大。每种路线分别记录软件版本、输入校验、输出记录数、空间峰值和下游验收。

流式转换并非所有场景的默认优选。若后续多次重复分析，保留原始FASTQ并建立受控索引或缓存，可能比每次解压或转换更合适；这要根据工具支持和I/O实际测量决定。咨询算力资源前先检查格式、工具和工作盘，不将gzip读取缓慢自动

等同于CPU不足。

把质量丢弃的决定记录给下游

FASTA输出应附一份转换记录，说明源FASTQ指纹、工具版本、转换命令、输入和输出记录数、质量字段被移除、read ID规则及校验结果。将记录随输出文件一起交给下游，避免后续研究者误以为FASTA来自原始测序或仍保留质量信息。若生成的是压缩FASTA，应确认下游工具识别的压缩格式和后缀一致。文件名不应覆盖原始FASTQ，也不要用含混的扩展名命名产物。

若下游需要随机区间访问，压缩FASTA可能还要建立索引；索引会产生额外文件并需要与序列版本配套。索引完成后抽取几个已知记录或区间，核对标题、序列和记录总量。计划清理转换中间文件之前，先确认下游文件和索引均可读取，且原始FASTQ仍按项目策略保留。

若输出记录数与输入不一致，先检查工具是否过滤或改写记录，再判断是否符合预先约定的转换目标。

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

seqtk官方标准输入与gzip序列处理 · 来源资料

GNU Bash pipefail · 来源资料

GNU gzip官方手册 · 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。