# 按官方要求整理GEX与ATAC文库并运行样例

完整操作教程 · 参考固定 · 往返恢复验收

核对同一Multiome实验的两类文库，用三列CSV检查器和ARC v2.2跑通联合输入。

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。

Cell Ranger ARC用于同一次Multiome实验获得的GEX和ATAC联合分析，不是把任意RNA样本与任意ATAC样本拼在一起。当前v2.2要求两种数据共同输入，不能用ARC只处理其中一类。先核对GEM well、文库与参考关系，再准备libraries CSV。

## 从建库记录确认配对关系

| 资料 | 需要确认 | 常见误用 |
| --- | --- | --- |
| GEM well与样本 | GEX/ATAC来自相应同一批细胞 | 拿不同实验的文库硬配对 |
| 解复用样本名 | 每种文库实际Sample_ID | 强行令文件前缀相同 |
| FASTQ目录 | 绝对路径及测序批次 | 一格逗号拼多个目录 |
| 参考 | ARC兼容包、组装及版本 | 以普通GEX参考目录代替 |
| 运行单元 | 每个GEM well独立count | 把生物学重复混成一次输入 |

同一文库多次测序可按官方规则增加行，使用对应library_type；不能将它们当独立生物学重复。CSV的sample用于选FASTQ，GEX与ATAC的Sample_ID可以不同，是否属于同一GEM well须由原始记录确认，软件路径检查不会替你证明实验关系。

## 建立三列文库文件

以下保存为libraries.csv。library_type区分大小写，必须使用Gene Expression与Chromatin Accessibility。目录、前缀都是示例，替换为真实交付；同一格fastqs不接受逗号分隔路径，跨目录需要增加相应行。

```
fastqs,sample,library_type
/absolute/fastqs/gex,Example_GEX,Gene Expression
/absolute/fastqs/atac,Example_ATAC,Chromatin Accessibility
```

先用标准库检查结构、路径、类型和重复条目，写出新的规范化路径版本。保存为check_arc_libraries.py，执行python check_arc_libraries.py libraries.csv libraries.checked.csv。它只检查文件表，不检查FASTQ配对内容、细胞身份或实验质量；源清单保留不改。

```
import csv, sys
from pathlib import Path

source, target = map(Path, sys.argv[1:3])
with source.open(newline="",encoding="utf-8-sig") as handle:
    reader = csv.DictReader(handle)
    assert reader.fieldnames == ["fastqs","sample","library_type"]
    rows = list(reader)
assert rows
allowed = {"Gene Expression","Chromatin Accessibility"}
seen = set()
for row in rows:
    assert all(row.get(k) for k in ("fastqs","sample","library_type"))
    assert row["library_type"] in allowed and "," not in row["fastqs"]
    folder = Path(row["fastqs"])
    assert folder.is_absolute() and folder.is_dir()
    row["fastqs"] = str(folder.resolve())
    key = (row["fastqs"],row["sample"],row["library_type"])
    assert key not in seen
    seen.add(key)
assert {x["library_type"] for x in rows} == allowed
with target.open("x",newline="",encoding="utf-8") as handle:
```

```
    writer = csv.DictWriter(handle,fieldnames=["fastqs","sample","library_type"])
    writer.writeheader()
    writer.writerows(rows)
print("validated rows:",len(rows))
```

## 在兼容环境运行一次新任务

ARC_REF需为已经验证的ARC参考；CORE和MEM变量填真实获配资源。以下Bash代码按当前v2.2显式保留BAM并关闭细胞注释，以固定示例范围。正式项目要使用注释时应按版本准备相应条件，不能将关闭功能后的结果当同一任务。

```
set -euo pipefail
: "${ARC_ID:?■■■■■■■ID}"
: "${ARC_REF:?■■ARC■■■■}"
: "${ARC_CORES:?■■■■■■■}"
: "${ARC_MEM_GB:?■■■■■■■■}"
case "$ARC_ID" in *[!A-Za-z0-9_-]*|"") exit 2;; esac
test "${#ARC_ID}" -le 64
test ! -e "$ARC_ID"
test -d "$ARC_REF" && test -r libraries.checked.csv || exit 2
cellranger-arc --version
cellranger-arc count --id="$ARC_ID" --reference="$ARC_REF" \
  --libraries=libraries.checked.csv --create-bam=true --disable-cell-annotation \
  --localcores="$ARC_CORES" --localmem="$ARC_MEM_GB" \
  > "$ARC_ID.stdout.txt" 2> "$ARC_ID.stderr.txt"
```

找不到FASTQ先核对sample前缀及目录，不盲目放宽到整个文件夹；参考错误先检查ARC兼容性；仅一类文库时应回到实验和流程选择。localcores/localmem是本地任务申请安排，不会突破平台额度，也不自动提供系统级隔离。

## 联合结果需要联合验收

先确认正常结束和web报告中的样本、两种模态输入规模，再核对GEX/ATAC各自质量、联合细胞识别、矩阵特征类型与barcode范围。ATAC片段和索引、峰、所需BAM及后续联合分析输入都应能访问；只看到一份矩阵不能代替整套输出验收。

不要为了得到更多细胞随意设置min-atac-count与min-gex-count。它们改变联合细胞识别，应按官方要求与报告一起评估。保存原始清单、规范化清单、参考、软件和命令，再把这一套有效单样本作为资源试跑基准，而非一次批量启动所有未核对的配对。

启动前还需核对受支持Linux与CPU指令集。官方v2.2通用基线为8核Intel/AMD且支持AVX、64 GB内存、1 TB可用盘，另列48核与320 GB推荐值；这些数值不是每个Multiome任务完成保证，仍应以代表性联合任务测量。

两类模态的质量口径不同：GEX的UMI、基因与ATAC片段及峰信号应分别检查，再看联合细胞范围，不把一个模态的高计数用作另一个模态质量合格的证据。

## 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


Cell Ranger ARC v2.2配对文库分析 · 来源资料

Cell Ranger ARC命令参数 · 来源资料

Cell Ranger ARC系统要求 · 来源资料

Cell Ranger ARC 官方 FASTQ 与 libraries CSV 输入说明 · 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。