检查路径、依赖、文件格式与平台差异

完整操作教程 · 参考固定 · 往返恢复验收

迁移到Linux前盘点输入路径、解释器、架构和文本格式，使用只读清单检查定位第一处不兼容。

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。

从Windows或Mac迁到Linux，代码逻辑相同也可能因路径、二进制和文本格式不同而失败。先准备一个能在原系统复跑的小样例，将代码、输入、版本和预期结果列清，再在Linux逐层检查，避免一次搬全部数据后才发现关键工具不支持。

先分清可复制与需重建的部分

源代码、配置和允许迁移的数据可以按清单复制；Windows的exe、Mac二进制及原系统包库一般不能直接当Linux程序使用。Python虚拟环境需在目标位置重建，Conda显式包清单通常只适合同一平台，R包中的编译部分也要按目标系统准备。

检查项	典型差异	处理方式
路径	盘符、反斜杠、大小写与空格	配置为明确的Linux路径或项目相对路径
软件	操作系统、CPU架构、GPU后端	按目标平台准备匹配构建
文本	CRLF、BOM、编码和分隔符	只对已确认文本做格式调整
运行入口	交互对象、默认工作目录	使用可重复的完整启动命令

Linux通常区分文件名大小写，Sample和sample可能是两个路径；原系统是否区分还取决于文件系统设置。包含空格的变量应加引号，不靠重命名掩盖样本对应。不要对FASTQ压缩包、BAM或模型二进制批量执行换行替换。

用样本清单检查实际输入

以下例子限定项目根目录下samples.csv，表头为sample,r1,r2，文件路径使用正斜杠相对写法。保存为check_paths.py，在根目录执行python check_paths.py；它检查路径与基本样本规则，不打开测序内容，也不改写文件。

```
import csv
from pathlib import Path, PurePosixPath

root = Path.cwd()
with (root / "samples.csv").open(encoding="utf-8-sig", newline="") as handle:
    rows = list(csv.DictReader(handle))
assert rows
ids = [row["sample"].strip() for row in rows]
assert all(ids) and len(ids) == len(set(ids))
for row in rows:
    paths = []
    for key in ("r1", "r2"):
        text = row[key].strip()
        relative = PurePosixPath(text)
        assert text and "\\" not in text and not relative.is_absolute()
        assert ".." not in relative.parts and ":" not in text
        path = root.joinpath(*relative.parts)
        assert path.is_file(), f"missing input: {text}"
        paths.append(path.resolve())
    assert paths[0] != paths[1], "R1 and R2 must differ"
print("samples", len(rows), "input_paths_ok")
```

示例清单可写一行S01,data/S01_R1.fastq.gz,data/S01_R2.fastq.gz。共享参考或外部挂载需要单独配置绝对路径，不应为了通过这个相对路径例子随意复制大型数据库。清单检查通过仍需校验数据内容和配对关系。

再检查脚本与依赖

在目标环境记录系统、架构、解释器及主工具版本。出现bad interpreter或脚本首行异常时，检查shebang、执行方式和文本换行；出现共享库、架构或导入错误时，回到对应软件构建。CSV读取异常先查编码、表头和分隔符，不直接把错误列填成空值。

只对受版本管理的文本副本规范编码与换行，保留变更差异。外部工具调用使用参数列表或正确引号，不拼接未经检查的样本名。原来依赖Excel手工处理的步骤，应把输入格式和导出设置写入说明。

迁移清单还应记录软链接实际目标和外部程序调用。复制了一个链接不代表目标文件也随之到达；路径看似正确而读取失败时，分别核对挂载、目标权限和内容。使用大小写不同的两个同名参考时，明确实际被加载的是哪一份，再解释分析差异。

最后才比较分析结果

先让同一小样例在Linux完整运行，再检查样本数、特征ID、关键数值和输出结构。路径通过、文件复制完成或主程序能打印版本，都不是科学结果已复现的证据。记录仍未迁移的依赖与资源条件，再决定是否扩大到正式数据。

平台迁移先做目录映射再复制数据

把源系统路径和目标Linux路径列成映射表，逐项注明输入、参考、软件、模型、临时盘、结果和归档位置。确认目标挂载点名称、大小写、读写权限、软链接目标和剩余配额；不要依赖Windows盘符或Mac外置盘路径。复制大型FASTQ、BAM和模型前先抽查源/目标文件大小和校验和，分批迁移并记录失败重传。原始数据只读保存，转换、解压和测试写入新的工作目录。

环境锁文件不等于跨操作系统二进制可用

Conda或Python环境文件可能包含平台专有包、构建号或绝对路径；Windows导出的环境不一定能在Linux直接创建成功。保存软件包名称与版本作为参考，在目标系统按官方安装说明重新解析环境，记录新生成的锁定信息。R包若包含本地编译代码，应在Linux环境构建并检查系统库依赖。预编译二进制需匹配目标CPU架构和系统库，不能把.exe或macOS二进制直接复制后仅改文件名。

换行、BOM和可执行位要一项项检查

Shell脚本从Windows传到Linux后可能含CRLF换行或BOM，导致shebang识别错误；CSV也可能因BOM、逗号/分号分隔符、编码或小数点地区设置而读错。迁移前确认编码、换行、分隔符和路径转义，目标机上检查file、head、脚本语法与权限。不要在Excel中打开并另存大样本清单而不复核，因为自动改写日期、科学计数法和分隔符会破坏样本ID或路径。

做一份只读的端到端小样本

选择一对已知FASTQ或小型公开数据，在目标系统从检查文件、读取配置、调用软件到写出预期文件走完整链路。验证文件内容和预期值，而不只是命令退出码；同时核对日志、输出位置、线程、数据库路径和输入校验和。小样本通过只证明基本迁移路径可工作，不代表真实规模的性能或科学结果可靠。完成验证后再复制其余大文件并提交正式任务。

将旧环境保留到新环境通过验收

迁移时不覆盖旧环境或唯一数据副本。记录迁移窗口、文件清单、校验和、软件安装日志和未解决依赖；新环境验证通过前，保留源路径或快照可回退。若项目中的结果受随机种子、浮点库或GPU实现影响，结果可能存在数值差别；设置可比较的验收标准，关注结构、字段和统计结论，而不是要求所有浮点字节完全一致。

符号链接和大小写要在目标机实测

跨平台文件夹可能包含指向原盘的符号链接或快捷方式。Windows快捷方式不是Linux软链接；已复制的软链接也可能指向源盘旧路径。运行清单前，逐个检查目标路径最终指向的位置与可读权限。Linux通常区分大小写，因此SampleA与samplea是不同目录；对样本ID重复或路径大小写不一致应停止并核实，而不是自动改名造成样本对应关系变化。

把软件版本检测放进小样本验收

迁移清单中列出可执行文件版本、R/Python版本、系统库、CPU架构、GPU驱动、数据库版本和容器镜像摘要。对每个关键工具执行帮助/版本信息，并在小样本上检查预期输出字段和基本统计。新环境能启动程序不代表格式、参数和结果文件结构与旧环境一致；升级造成输出差异时，先做版本对照再提交大批数据。

编码和分隔符检查要考虑表格软件

CSV/TSV用逗号、制表符或本地分隔符的差异会影响脚本读取；中文路径、空格和引号也可能在shell中被重新解释。把原始清单作为只读文本保存，再用脚本输出规范化后的目标清单并人工抽查几行。不要手工在多个软件之间另存来回，否则可能引入BOM、日期重写或自动科学计数法，导致路径和样本ID静默变化。

迁移完成后的回退与责任边界

每个项目指定旧数据的保留期限、新服务器管理员、环境维护人及迁移失败时的回退步骤。迁移结束并通过小样本验收后，再逐批转入正式数据；不要让组内成员各自复制出多个未经核对的版本。汇报迁移结果时把“文件复制完成”“环境可运行”“样本级结果一致性已核查”分开描述，清楚说明尚未验证的项目。

容器化可减少环境漂移但不消除路径问题

容器镜像能固定系统依赖，但宿主机GPU驱动、设备映射、文件权限和挂载路径仍会影响任务。保存镜像摘要而非只保存tag，确认镜像架构与节点CPU/GPU匹配。输入、数据库、模型和输出应分别挂载；容器内的临时目录可能落在默认overlay空间，若其容量有限，应按软件支持方式映射到高速工作盘。

迁移前后样本表应能逐行比对

目标环境上重新生成规范化的样本路径表，保留源路径、目标路径、样本ID、校验和和检查结果。对比两份清单的样本数、ID顺序和每个文件校验和，发现缺失时先修复复制，不要让流程自动跳过缺失样本。样本清单作为任务输入版本保存，后续有新增样本时另建清单版本，避免旧任务被无意改变。

检查权限、ACL和挂载选项

迁移后文件可见不代表任务用户有权限读取。检查目录遍历权限、文件读取权限、组权限、ACL、NFS挂载策略及只读挂载；容器用户的UID/GID也可能与宿主机不同。对输出目录做一次受控创建与删除测试，确认不是所有人可写的公共路径。权限配置按课题组最小需要授权，避免为了让脚本跑通就递归开放所有文件。

完整性核对要按文件类型安排

普通文件可用SHA256核对；稀疏文件、符号链接、数据库目录和容器镜像需要确认对应校验方式。压缩文件校验只能证明字节一致，不保证解压后的内容格式能被应用读取。对数据库和HDF5/BAM等结构化文件，应在目标端使用相应工具进行只读完整性检查。保留清单、校验结果和失败项，方便迁移后复核。

适用范围与结果归档

目标环境的CPU/GPU架构可能与源环境不同，尤其影响编译包、ROOT依赖、CUDA及HDF5等本地库。记录实际架构与依赖版本，并在迁移小样本中验证读写。若软件只能在某架构运行，应在任务规划时明确节点类型。

迁移总结中列出未成功的样本或工具、错误信息、是否已重试以及当前数据所在位置。不要把迁移任务标记为全部完成，直到文件校验、环境运行和目标用户读取都分别有记录。

旧系统和新系统可使用同一个小样本清单做输入验证，但版本差异应记录在报告中。对于文件格式转换，保留转换前后校验和、转换命令和字段统计；只有内容层面的核对通过后，才将目标文件作为后续分析输入。

小结

网络挂载下若脚本报告文件存在但实际读取失败，检查目标机是否挂载了正确文件系统、挂载是否只读或临时断开。路径存在不等于内容可读；可在作业启动前逐个打开关键输入并记录校验和，失败时停止提交，避免生成部分结果。

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

Python：路径的平台语义 · 来源资料

Python：虚拟环境需要重建 · 来源资料

Conda：显式环境清单的平台边界 · 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。