整理包与系统依赖，建立可核验的离线安装清单

完整操作教程 · 参考固定 · 往返恢复验收

离线服务器安装Bioconductor包，关键不是单独拷贝一个tar.gz，而是准备与目标R、Bioconductor、操作系统和架构相匹配的完整依赖集合，并在隔离环境演练。

运行边界：目标R/Bioconductor与OS需匹配；离线包和系统依赖清单未在此服务器验收。

先固定目标环境和版本

离线安装失败通常不是“网络断了”这么简单，而是准备包与目标系统不匹配。Bioconductor发行版与R版本有对应关系，发布周期也与CRAN不同；同名包在release和devel仓库中的版本可能不一致。先确认服务器的R版本、Bioconductor版本、操作系统、CPU架构、编译器和系统库，再决定要安装哪批包。不要在联网电脑上用最新R下载包，然后直接复制到较旧服务器上。

在目标R环境中检查`BiocManager::version()`、`BiocManager::repositories()`和`BiocManager::valid()`，把输出存入安装清单。Bioconductor官方推荐用BiocManager管理其包及仓库，并提供valid()检查包集合是否混用了不匹配版本。若目标包包含C/C++/Fortran代码，离线环境还需要兼容的编译器、头文件和系统库，单有R源码包不能保证安装成功。

明确包来源：哪些来自CRAN、Bioconductor软件仓库、注释仓库、GitHub或实验室私有仓库。记录包名、版本、源仓库、依赖关系和SHA256。GitHub包最好记录release或commit，而不只是仓库名称。私有仓库凭证不要写进清单或镜像；需要在联网准备端通过受控凭证获取，并按机构规则转移安装包。

准备项	必须记录	常见缺漏
R与Bioconductor	精确版本、平台、架构	联网端与目标端版本不同
R包集合	版本、来源、依赖	只复制顶层包
系统依赖	编译器、头文件、共享库	安装时报缺.so或编译器
仓库元数据	PACKAGES索引、文件摘要	有tar.gz但索引缺失

在联网端解析完整依赖闭包

联网端应尽量与目标服务器使用相同R版本、操作系统家族和架构。先建立干净的项目library或隔离容器，再按目标Bioconductor版本安装所需顶层包，让包管理器解析所有传递依赖。记录最终已安装包列表、版本和仓库来源，并把依赖集合导出为可核验清单。若直接从已经使用多年的个人library打包，里面可能混有旧版本、开发版和手工安装包，难以判断哪些是真正需要的依赖。

CRAN和Bioconductor包可按平台下载二进制或源码包。离线目标若没有编译器，准备源码包可能无法安装；二进制包又需与R、操作系统和架构匹配。编译源码时还需系统依赖和可用构建工具。对复杂包，可优先在匹配的离线镜像或容器中预构建环境，而不是把所有源码包当作可直接使用的安装包。

本地R仓库通常需要按R仓库目录结构放置包文件并生成PACKAGES索引；对于不同仓库类型，应分别保存来源和索引。下载动作本身也需验收：包文件是否完整、文件名和版本是否符合清单、checksum是否匹配、同名不同版本是否冲突。不要把包缓存目录当成完整离线仓库，因为缓存可能只含曾经下载过的部分包，且来源映射不一定保留。

把包与仓库元数据一起传入隔离区

将安装包和索引打成只读交付包，附上生成日期、目标环境、包清单、来源和SHA256。通过单位批准的介质传输，进入离线区后再校验签名或摘要。原始安装包只读保存一份，实际安装使用副本。若有多个目标服务器，确认系统补丁、R版本和架构一致；不一致时要维护多个仓库快照，不能让一个仓库混放未经验证的二进制包。

在离线区将本地仓库加入R的repos配置，并先检查能否读取PACKAGES索引和包元数据。安装时明确使用目标Bioconductor release对应的本地软件、注释和数据仓库；如果依赖解析尝试访问外网，说明仓库设置或依赖包仍有缺项。安装日志要保留warning、下载地址和失败依赖。不要在正式用户library里反复尝试不同版本，以免破坏已有项目。

在空library中做断网验收

真正的离线验收是在网络被禁用或离线容器中，从空library开始安装顶层包，再运行BiocManager::valid()或包自身的依赖检查。随后执行一个小型功能测试，例如加载包、调用一个不会下载外部数据的核心函数、检查系统共享库链接。若安装需要AnnotationHub或外部参考数据，必须另外准备该数据资源和缓存；“R包能加载”不代表完整分析流程已经离线。

验收还要测安装失败和回滚：缺少一个依赖时，错误信息是否指向清单中未准备的包；安装中断后是否留下半安装目录；是否能删除隔离library并从清单重建。确认通过后，保留仓库快照、安装日志、R sessionInfo、包校验和和功能测试结果。任何包升级都会改变依赖闭包，应生成新仓库版本并重复验收，不能在原仓库中覆盖包文件。

构建离线仓库时处理源码、二进制与平台差异

Bioconductor和CRAN仓库可以提供源码包，也可能为特定平台提供二进制包。Windows、macOS和Linux的包格式及系统依赖不同；同为Linux也会受发行版、glibc和编译器影响。先决定离线服务器是否允许编译源码。若允许，准备与目标系统匹配的编译器、make、Fortran工具链和包所需系统开发库；若不允许，必须准备适配目标R版本和操作系统的二进制包。不要把一台联网工作站的个人包库目录原样复制到服务器，它可能含有符号链接、缓存路径和本机编译产物。

构建仓库时按实际R仓库布局放置包，并为每个平台分别生成索引。R包可使用`tools::write_PACKAGES()`生成本地仓库索引，之后用与目标环境相同的R版本检查能否读取。仓库里保留源包不代表R会自动选择正确版本；R版本、Bioconductor release和平台共同决定依赖解析。安装日志中若仍出现外网URL，说明repos设置没有完全指向本地仓库或某个依赖来源尚未准备。可在隔离环境阻断网络后重复测试，确认没有隐式下载。

包文件传入离线区前生成清单和SHA256。接收端逐一计算摘要并比对，防止文件传输截断或同名包被覆盖。对每个仓库版本保留独立目录，目录名含R、Bioconductor和平台标识；不要将新版本的PACKAGES索引覆盖到旧包文件上。若使用只读介质，将构建工作区与发布仓库分开：先在可写暂存目录构建和测试，审核通过后复制到只读发布区。

顶层包的传递依赖与注释资源需分开清点

安装一个Bioconductor顶层包可能需要CRAN包、Bioconductor基础包、注释包、实验数据包以及系统库。仅按DESCRIPTION文件手动列出直接依赖，容易遗漏Imports、LinkingTo、Suggests中的测试依赖或平台条件依赖。应在目标版本的联网环境中完成安装并记录最终library包集，再按目标用途决定是否包含Suggests。若应用代码会调用这些可选包，测试集合中必须包含它们，即使核心库安装本身不依赖。

注释和参考数据与R包不是一回事。AnnotationHub、ExperimentHub或其他数据包可能在首次调用时下载大型资源，资源还可能因版本或记录ID变化而更新。对离线流程，需要单独确定数据文件、版本、许可、摘要和访问路径，并测试程序是否会偷偷访问远程缓存。一个包成功安装，但流程启动时仍尝试联网下载数据，整体仍不具备离线能力。

若使用离线仓库镜像服务，确认它镜像的是release还是devel，镜像时间和同步策略是什么。复制最新版索引但旧包文件可能产生“索引有、文件无”；只复制文件而漏掉PACKAGES索引又可能使install.packages找不到包。每次同步后用清单比较文件数、版本和摘要，并在新library中进行依赖闭包安装。

处理无法满足的包依赖

如果某个包无法构建，先确认失败点是R层依赖还是系统层依赖。查看安装日志和`configure`输出，定位缺失头文件、共享库、编译器或外部程序。系统管理员应在隔离节点或容器里验证依赖，不要由普通用户直接修改全局系统库。对于已有二进制包，检查其目标R版本、架构和系统ABI；无法确认兼容时，应在目标平台重新构建。

有些旧项目依赖的包版本已从当前仓库移除。可以从授权的包归档、机构镜像或项目维护者仓库恢复，并记录来源与许可证；不得从随机网站下载未知二进制。若无合法来源，和PI确认是否能改用可替代实现，并把方法变更记录为新版本，而

不是伪装成原环境恢复成功。

用户私有library之间不应相互混用包版本。共享离线仓库可以只读提供包文件，各项目仍在独立library中安装；这能降低一个项目升级影响其他组的风险。准备好锁文件或包清单后，先用空library安装，不要在管理员维护的共享R库中做实验。失败时清理的也是专属测试library，不要递归删除全机的renv或R缓存。

离线验收要覆盖真实入口和软件升级

从全新账号或临时用户开始，验证R能访问本地repo、安装指定包、加载包并运行最小功能。确保安装过程中没有网络请求，并且仓库包的hash与manifest一致。接着运行一个小型、无敏感数据的分析脚本，检查读取、核心函数、保存输出和重新读取。保存控制台日志、sessionInfo、BiocManager::valid()结果和系统依赖清单。

若目标环境将来会升级R或Bioconductor，应创建新的仓库快照，而不是直接覆盖已验收版本。升级后单独跑兼容性检查，保留旧版本供历史项目恢复。编译器或系统补丁更新也可能需要重新编译包；一次离线安装通过不代表永久兼容。团队应指定包仓库负责人、版本冻结日期和更新审批流程。

示例：对离线包仓库做清单校验

这组shell命令只盘点包文件和索引，不执行安装，也不假设仓库一定兼容。将目录改为经批准的离线仓库副本；检查摘要文件是否由可信联网端生成。之后仍须在目标R中测试索引和依赖解析。

```
set -euo pipefail
REPO="${REPO:?set approved local repository path}"
find "$REPO" -maxdepth 3 -type f -printf '%P\t%s bytes\n' | sort
find "$REPO" -name SHA256SUMS -print -exec sha256sum -c {} \
  --directory "$REPO"
# In the target R environment, verify repository metadata and package versions.
```

GNU sha256sum不同版本对`--directory`选项支持不同，实际使用前查看`sha256sum --help`。更通用的做法是在仓库根目录执行清单校验，或用Python逐条检查相对路径。不要用shell通配符遍历未经审核的包名运行安装。

离线R仓库还应包含可读取的PACKAGES索引和一致的目录层次。一个清单中有文件不等于R包管理器会发现它。将仓库复制到目标区后，以同一套R和Bioconductor版本运行metadata查询、空library安装和最小功能测试，最后把全部日志及仓库摘要归档。若无法满足任何一步，应明确标出未验证部分。

外部数据和系统组件纳入离线维护计划

如果分析包首次运行时下载注释或参考数据，单独制作数据清单、版本和校验摘要，并确认其许可允许在离线服务器使用。包仓库更新与数据资源更新分开审批；包版本不变而数据库版本改变，也可能改变分析结果。离线区的安装维护由指定人员负责，新增包先在测试仓库验收再发布，避免多人把未经测试的tar.gz直接放进共享目录。

完成安装后冻结仓库快照并写入README，说明适用R、Bioconductor、操作系统和架构。用户项目引用该快照路径并记录摘要；升级时复制到新版本目录，让旧项目继续使用旧仓库。这样能在网络策略变化或公共仓库更新后，仍明确知道当时装的是什么。

配套练习包

本PDF对应的ZIP包含本文代码、主题模板和SHA256校验文件。运行前核对包内清单，并按文章说明在独立目录验证。

参考资料

Bioconductor安装与版本说明 · 来源资料

Bioconductor Docker环境 · 来源资料

renv package sources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。