scRepertoire多样本流程：整理链信息、样本标识和计算记录

完整操作教程 · 参考固定 · 往返恢复验收

按文库读取10x TCR contig表，固定过滤与克隆定义，用样本前缀消除barcode碰撞并保留原始对应。

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。

多样本VDJ数据合并，最先要解决的是身份可追溯。同一个10x barcode可能出现在不同文库，不能只按裸barcode合并，也不能看到跨供者相同CDR3就认定同一个物理克隆。本篇限定兼容10x输入的TCRαβ，BCR及其他受体按相应入口单独处理。

先整理样本与文库清单

建立vdj-samples.csv，表头sample,contigs，每行是稳定样本ID和实际filtered_contig_annotations.csv路径。另保留供者、时间点、组织、文库与GEX对应关系。一个细胞可有多条contig，因此contig行数不等于细胞数；不要为避免重复barcode把合法多链记录直接去重。

身份层级	保留什么	避免混淆
样本/文库	稳定sample及来源路径	同名barcode跨文库碰撞
细胞	原barcode和样本前缀后ID	任意去掉-1或改写后缀
链	contig、TRA/TRB、序列与生产性	多链不自动当双细胞
克隆定义	nt、aa、基因或strict及链规则	不同定义的克隆数直接比较

使用当前参数名组合TCR

下面按官方scRepertoire2.7.2页面的点号参数名运行，旧版不支持时先核对版本，不把参数被忽略当默认正确。示例保留单链及多链，不按最高表达挑两条，同时明确过滤非生产性链；这些是需要项目确认的方法设置。

```
library(scRepertoire)
meta <- read.csv("vdj-samples.csv", stringsAsFactors=FALSE)
stopifnot(nrow(meta) > 0, !anyNA(meta[, c("sample","contigs")]))
stopifnot(all(grepl("^[A-Za-z0-9_-]+$", meta$sample)))
stopifnot(!anyDuplicated(meta$sample), !anyDuplicated(meta$contigs))
stopifnot(all(file.exists(meta$contigs)))
needed <- c("remove.na","remove.multi","filter.multi","filter.nonproductive")
stopifnot(all(needed %in% names(formals(combineTCR))))
contigs <- lapply(meta$contigs, function(path) {
    x <- read.csv(path, stringsAsFactors=FALSE)
    stopifnot(all(c("barcode","chain","contig_id","cdr3","cdr3_nt","productive") %in%
names(x)))
    stopifnot(nrow(x) > 0, !anyDuplicated(x$contig_id))
    stopifnot(all(x$chain %in% c("TRA","TRB")))
    x
})
names(contigs) <- meta$sample
combined <- combineTCR(contigs, samples=meta$sample,
                       remove.na=FALSE, remove.multi=FALSE,
                       filter.multi=FALSE, filter.nonproductive=TRUE)
barcodes <- unlist(lapply(combined, function(x) x$barcode), use.names=FALSE)
stopifnot(length(combined) == nrow(meta), length(barcodes) > 0,
          !anyDuplicated(barcodes))
stopifnot(!dir.exists("vdj-run"), dir.create("vdj-run"))
saveRDS(combined, "vdj-run/combined-tcr.rds")
write.csv(meta, "vdj-run/input-map.csv", row.names=FALSE)
write.csv(data.frame(sample=meta$sample,
                     input_contigs=vapply(contigs, nrow, integer(1)),
                     output_cells=vapply(combined, nrow, integer(1))),
          "vdj-run/count-audit.csv", row.names=FALSE)
```

```
writeLines(capture.output(sessionInfo()), "vdj-run/session.txt")
```

样本列和前缀会影响后续与Seurat或SCE连接；先检查实际combined中的barcode，再按明确对应表对齐GEX细胞。不要靠全局替换下划线或删除后缀来让两边“看起来匹配”，尤其同一供者多个时间点和不同文库时。

验收过滤与克隆含义

逐样本比较输入contig数、有效细胞数、单链、多链和非生产性过滤，任何样本消失都应有解释。若没有可用TCR，也应登记为空结果或排除依据，不能悄悄丢掉后仍说全部样本成功。保留原表，便于回查过滤改变。

CTnt、CTaa、CTgene与CTstrict对应不同定义，分析前固定需要的链和规则。共享氨基酸序列不保证相同核苷酸、抗原特异性或共同细胞来源；跨供者比较尤其不能把序列相同直接当谱系追踪证据。

队列扩展时保持同一处理协议

先用少量已知样本确认ID、过滤和输出，再批量读入；样本很多时参考表与中间对象也占内存。分批保存输入映射及完整组合对象，汇总时保持同一克隆定义，不重复导入同一文库。需要全队列重定义克隆的步骤不能由分批标签随意替代。

读入错误先查CSV格式和版本，链类型不符先选择合适方法，ID碰撞先修清单。保留失败样本和错误日志，以新目录重试；代码、包版本或过滤规则变化后重新核对统计，不能只比较最终图形是否相似。

队列汇总前保留一张样本与临床分组的映射表，输入样本名和后续细胞对象的样本字段逐项对照。若同一barcode在不同文库出现，必须先加准确样本前缀；不要按全队列去重后只保留第一条，这会误删真实细胞并影响克隆比例的分母。

本例限定含productive列的标准10x注释表；缺少该列就拒绝输入，避免API无法按此列过滤却被误报为已去除非生产性链。

输入表和barcode的组织方式

样本清单中的每条记录应表示一个独立VDJ文库，而不是一个供者或组织大类。若一个样本有多个文库，sample列要体现文库ID，并另建映射表记录donor、时间点、组织、GEX文库及处理条件。路径最好固定到项目内相对路径，迁移到服务器后再做文件存在、散列和权限核验。contig文件可能包含多条链及不生产性contig，不能简单按barcode去重，否则会丢掉后续判断所需证据。

scRepertoire在组合时会对barcode添加样本标识，便于避免不同文库的裸barcode碰撞。运行前检查样本名稳定、唯一且仅含预期字符；保留原始barcode、样本前缀后barcode和contig_id之间的关系。若与GEX对象合并，需确认前缀格式一致，并先用小批细胞做键匹配，避免组合后出现大量没有VDJ或没有表达信息的细胞。

productive、多链和克隆定义的选择

过滤非生产性contig是分析定义的一部分，应保留原始输入和过滤前后数量。多链细胞可能是双细胞，也可能有生物学上可解释的链组合；不能只保留最高表达链而不记录规则。排查时统计每个细胞的TRA/TRB链数、生产性状态、序列完整性和重复contig_id，结合实验设计判定异常，而不是将所有多链都自动当作双细胞。

克隆型可以按氨基酸、核苷酸、V/J基因、链配对或strict定义。不同选择回答的问题不同，跨时间点或样本比较必须保持同一定义，并报告软件版本和参数。相同CDR3序列本身不证明细胞来自同一个物理克隆，也不直接证明抗原特异性；若研究涉及谱系关系或抗原结合，应结合额外证据和相应验证实验。

结果检查与下游整合

组合结果后核对每个输入样本文库是否都有对应输出，比较输入contig数、生产性链数、输出细胞数和过滤比例。比例异常时回看上游Cell Ranger版本、V/D/J参考、contig注释文件与productive列含义。将VDJ结果与GEX整合时，先确认细胞条形码匹配率及每个样本的重复键；匹配率低时先查前缀和后缀，不要直接扩大模糊匹配规则。

克隆扩增分析应在样本、时间点和供者层面保留结构，避免在合并时丢掉生物学重复。一个大克隆可能受文库深度影响，比较时报告分母、克隆定义和归一化方式。结果表中区分原始克隆标识和跨样本重新定义的联合标识，不能把工具自动产

生的clone列当成跨文库已校正的谱系追踪结论。

批量任务和资源记录

VDJ contig表通常比完整表达矩阵小，但多样本对象及其元数据仍会占用内存。先单独组合一个代表性文库，再扩到整批；记录每样本contig行数、细胞数、峰值内存和运行时间。若任务主要在读写CSV阶段耗时，增加CPU可能无明显帮助；若输入对象累积在内存里，可评估逐文库处理并保存中间结果，但必须保证后续合并规则一致。

输出目录中建议保存合并对象、样本映射、计数审计表、sessionInfo、参数清单和日志。失败文库用新尝试目录重跑，不覆盖成功样本的结果。仅在所有输入文库都有明确状态且细胞键通过检查后，才生成最终汇总表。对含供者或临床信息的映射文件设置恰当访问权限，公开练习材料不要放入真实样本标识。

比较克隆时避免深度偏差

测序深度和有效VDJ细胞数会影响观察到的克隆数及扩增比例。跨时间点比较时，报告每样本可用细胞数和过滤比例，必要时按预先定义的下采样或统计方案做敏感性分析。下采样本身会损失信息，因此保留全量结果作为主数据，并将敏感性分析单独标注。不能把克隆数较多直接解释为免疫反应更强。

检查每个克隆的细胞组成与链证据，至少确认样本前缀、CDR3序列和链配对规则。相同克隆标签可能源于算法在当前定义下的归组，跨样本共享标签前要确认条形码和样本边界没有丢失。为图形展示重命名克隆时保留原始clone ID映射表，避免图中标签无法回查到输入contig。

故障恢复与输出状态

如果某个样本在组合阶段失败，不要重新运行全部样本并覆盖成功结果。先确认错误来自特定contig文件还是样本清单，再在独立尝试目录重跑该文库。成功输出应同时具备对象、样本映射、计数审计和sessionInfo；若只有部分文件存在，状态仍应是failed或incomplete。批次脚本须逐样本记录退出码，最后检查每个manifest行都有一个对应状态。

样本数很多时，分批运行可降低单次对象体量，但需固定scRepertoire版本、过滤参数和克隆定义。跨批对象合并前用一致的输入结构回归测试，并核对样本列和细胞ID没有在序列化/反序列化时变化。若版本升级，抽取固定样本做新旧版本结果对照，把差异纳入方法审阅后再处理全部数据。

一条从输入到交付的最小路线

先选一个代表性文库核对contig字段、样本标识和productive状态，再使用固定参数执行combineTCR。检查输出细胞条数、链组合和barcode前缀后，再扩到其他文库。第二阶段才把VDJ结果与GEX对象按稳定键合并。这样可以把字段不匹配、样本名错误和GEX前缀问题分开定位，避免一次大批量失败后难以判断是哪类错误。

正式交付包含输入样本清单、过滤参数、软件版本、合并对象、每样本计数审计、GEX匹配率和方法说明。若某些细胞没有受体信息，保留缺失状态并报告，不要把没有VDJ的细胞自动删除后再计算匹配比例。每次过滤都保留前后计数，让读者看见数据覆盖和选择过程。

多用户服务器上的文件安排

同一项目可将原始contig文件设为只读，将输出与缓存按用户或运行ID隔离。多人共享相同的CSV或RDS时，避免同时写入同一结果文件；先用暂存路径完成并核验，再移动到约定的只读结果目录。大型样本列表不要嵌入shell命令，放入受控manifest并校验路径，避免命令行截断或特殊字符导致输入错位。

VDJ链和克隆序列可能包含可识别的个体信息。按项目授权管理读写权限，限制临床映射表的可见范围，分享教程时只使用占位符或合成barcode。服务器环境应能说明项目目录、备份策略和清理规则，但最终的数据共享审批由研究团队和机构负责。

多样本审阅可以抽取每个样本的代表contig检查chain、productive、CDR3和barcode前缀，另抽查多链细胞确认过滤策略符合设计。抽样是快速发现字段问题的方式，不替代全量唯一性和覆盖检查。

正式生成克隆汇总前，锁定过滤规则和clone definition，保存一份参数快照。后续发现规则需要调整时新建结果版本，并说明影响了哪些样本和克隆计数。不要用新规则覆盖旧表而不保留差异。

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

scRepertoire官方：组合contig与克隆定义 · 来源资料
combineTCR当前参数与ID前缀 · 来源资料
scRepertoire当前组合实现 · 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。