SingleR多参考运行教程：参考复用、分批输入与结果核对

完整操作教程 · 参考固定 · 往返恢复验收

在同一进程训练多个固定参考，分块查询并保存完整分数与细胞顺序，区分计算分块和全体质量判定。

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。

多个SingleR参考同时用于大型查询，先确认物种、基因ID和标签词汇是否可比较。把两个参考直接拼起来，可能混淆批次与命名差异。本文采用当前支持多参考的训练和分类接口，参考固定一次训练，查询按细胞列分块，输出保留参考来源。

先固定表达与标签输入

references.rds为有名称的参考列表，各元素是含logcounts和label.main的SummarizedExperiment；query.rds是基因×细胞的已核验表达矩阵。参考表达需按官方要求标准化，基因ID体系一致，不自动去重或把不同命名硬凑成交集。

输入	核对	资源含义
参考列表	来源、版本、标签、表达及基因	读取与训练本身占用RAM
查询矩阵	基因行名与唯一细胞列名	RDS仍需完整读入
分块计划	固定全部基因及细胞列边界	只减少部分计算中间量
输出	细胞、标签、参考索引及完整结果	不丢掉不确定性信息

同进程训练一次，再分类各块

下面按已核对的当前源码接口使用num.threads；旧安装版不具备该接口时先查对应手册，不套弃用的并行参数。1000列只是教学分块值，按实测修改。每块保持相同基因集合与顺序，不能各自挑选不同高变基因。

```
library(SingleR)
library(SummarizedExperiment)
stopifnot("num.threads" %in% names(formals(trainSingleR)))
stopifnot("num.threads" %in% names(formals(classifySingleR)))
refs <- readRDS("references.rds")
query <- readRDS("query.rds")
stopifnot(length(refs) >= 2, !is.null(names(refs)), !anyNA(names(refs)),
          all(nzchar(names(refs))), !anyDuplicated(names(refs)))
stopifnot(ncol(query) > 0, !is.null(colnames(query)), !anyNA(colnames(query)),
          all(nzchar(colnames(query))), !anyDuplicated(colnames(query)))
stopifnot(!is.null(rownames(query)), !anyNA(rownames(query)),
          all(nzchar(rownames(query))), !anyDuplicated(rownames(query)))
labels <- lapply(refs, function(ref) {
    stopifnot("logcounts" %in% assayNames(ref))
    stopifnot(length(ref$label.main) == ncol(ref), !anyNA(ref$label.main))
    ref$label.main
})
stopifnot(!dir.exists("singler-run"), dir.create("singler-run"))
set.seed(101)
trained <- trainSingleR(ref=refs, labels=labels, assay.type="logcounts",
                        test.genes=rownames(query), num.threads=1)
groups <- split(seq_len(ncol(query)), ceiling(seq_len(ncol(query))/1000))
for (i in seq_along(groups)) {
    index <- groups[[i]]
    pred <- classifySingleR(query[, index, drop=FALSE], trained=trained,
                            prune=FALSE, num.threads=1)
    stopifnot(identical(rownames(pred), colnames(query)[index]))
    saveRDS(pred, sprintf("singler-run/block-%03d.rds", i))
    write.csv(data.frame(cell=rownames(pred), label=pred$labels,
                         reference=pred$reference),
              sprintf("singler-run/block-%03d.csv", i), row.names=FALSE)
}
writeLines(names(refs), "singler-run/reference-order.txt")
```

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

本例复用训练结果只限同一R进程。当前训练对象含不可直接序列化的C++索引指针，不把saveRDS(trained)当可靠持久恢复；跨进程应按版本支持方式重建并验证，保留参考、参数和基因清单更重要。

分块不等于自动保持所有判定一致

这里显式关闭prune，避免在每个小块内独立估计质量阈值后直接混为全体判定。它不意味着所有标签都可信；需用完整结果按统一规则检查差值、歧义和项目标记。若要剪枝，按安装版支持方法在正确全体范围重新评价，不将原始labels当最终验收。

先用小查询比较整批与分块的细胞、标签及参考来源，检查顺序变化和边界。标签词汇不同的参考即使给出同名结果，也不一定表示相同生物定义；结合各参考覆盖和上下文解释，不用简单多数投票替代方法设计。

按阶段定位内存与失败

分别记录参考读入、训练、每块分类和保存的时间与RAM。查询RDS本身过大时，分块索引不能消除完整读入开销，应另用已验证的磁盘支持表示；同时启动多个R进程还会重复加载参考，可能反而增加总内存。

报基因缺失或索引不匹配先查test.genes与实际查询，空块或重复细胞先查清单。失败保留本次目录，用新尝试重建，不能凭部分CSV存在判定整批完成。最终按全部输入细胞逐项核对，保留原始完整结果和统一的质量结论后再交付。

分块清单还应保存首末细胞标识、细胞数与完成状态。合并后逐项检查是否恰好覆盖原细胞一次；有一块失败就标为未完成，不把缺失细胞静默当成低质量样本删除。

参考之间的差异要先梳理

多个参考不一定采用相同组织、实验平台、基因命名、预处理和标签层级。训练前把每个参考的来源、物种、assay、标签列及标签数量整理成表，确认目标查询和参考共享足够的基因。若合并不同参考标签，需保留原参考名，不能把不同体系中同名但定义不同的细胞类型误认为完全一致。

先在小查询上分别对各参考分类，观察覆盖范围、标签分布和差异；若不同参考给出冲突结果，回到参考适用范围和表达矩阵处理方式排查。将多个参考结果并列比简单多数投票更透明。对于只在一个参考中出现的稀有标签，确认其支持基因、细胞数量和参考注释来源后再解释。

训练对象的作用是复用参考的索引和预处理逻辑。某些版本依赖外部指针等内部状态，不应假定可通过RDS序列化后跨进程恢复。本文示例只在同一R进程内先训练再分类；如需跨任务复用，应查当前版本官方支持的持久化方式，或以相同输入和版本重建训练对象并验收结果。

分块前后如何保证细胞不丢不重

按列分块主要限制分类阶段同时处理的查询细胞数，但示例先将整个query读入内存，参考也仍然驻留，所以这不是磁盘流式分析。切块大小应由小样本实测决定：块太大会增加峰值内存，块太小会增加循环、文件和日志开销。固定块大小、细胞顺序和参考集合，才能比较运行表现。

给每个块保存块号、起止位置、细胞数、首末细胞ID、运行参数、退出状态和文件散列。合并前把原始查询ID列表与所有块输出合起来，检查总数相等、没有重复、没有遗漏且标签行名顺序明确。任一块失败时将整个任务标记为未完成，不能仅凭其他块已生成CSV就交付部分结果。

不同块单独做剪枝时，阈值可能受块内组成影响。若计划剪枝，先确认所用SingleR版本的支持接口与阈值估计范围；不要把每个小块的独立质量判定直接拼成全体一致结论。可以先保存未经剪枝的分数与原始标签，汇总全部块后按统一规则评估不确定性，并保留筛选规则及未定标签。

控制线程与内存的实测方法

参考训练通常占用明显内存，分类阶段也会产生临时对象。记录读取每个参考、训练、分类每块和写出文件时的时间与RSS。R的对象内存估算函数不等于进程峰值，也不能看到单个矩阵大小就忽略多个参考及索引。使用共享服务器时，查询管理员确认作业内存和CPU额度，避免误把主机总资源当成当前作业可用资源。

并行度不要只依据机器逻辑CPU数设置。num.threads控制的范围应按当前安装版文档确认，同时考虑BLAS线程、外层任务和其他用户作业；过度并行可能导致内存相乘。先用单线程单任务建立基线，再在受控样本上测试是否提高线程确实缩短时间且没有增加错误或改变结果。

输入基因集合变化会影响训练和分类结果。若因内存问题减少基因或改用不同assay，记录明确的基因清单和矩阵处理方式，重新计算结果并检查标签稳定性。这种改变不是纯粹的资源设置，不应和相同方法下的不同服务器运行直接混为一组。

建议输出哪些文件，如何合并

每个查询块保存未经改写的SingleR预测对象和整洁CSV，另外写块清单记录块序号、查询细胞ID散列、参考名称、运行时间和R版本。合并应以细胞ID连接而非简单按行追加，并对输入细胞ID做集合比较：预期输入中每个ID应恰好出现一次，预测中额外出现的ID也必须为零。该检查可以防止上游索引变化导致结果行数正确、标签却整体错位。

如果保留多参考各自结果，表格应显式包含reference列和各自的label、score或delta信息。之后的综合注释可由统一规则生成一列最终标签，但不要丢弃原始参考输出。无法达成一致的细胞可以保留为ambiguous或待审，而不是强行多数表决；这种不确定性对稀有细胞尤其重要。

何时应考虑增加资源，何时先修方法

若同一查询在一个参考、单线程时读取阶段就失败，先检查参考文件、R内存限制和对象结构；若训练成功而只在较大分类块失败，可以在方法条件不变时逐步减小块大小，并复测结果是否完整一致。若训练本身占用主要内存，单纯缩小分类块帮助有限，因为参考和索引仍需保留。将失败发生阶段记录清楚，才能判断需要更多RAM、减少并发还是修复输入。

只有在正确读取、版本兼容、有效基因集合和测试样例都已通过之后，才基于实际峰值评估资源增加。若换了参考或减少了基因，结果不可直接与原结果混用；要重新确认细胞标签稳定性。计算资源决定能否在约定时间内完成，不会自动提升参考质量、纠正标签体系或证明细胞类型，因此资源决策和科学验收应分成两栏记录。

建立参考适用性检查清单

每个参考先回答：与查询是否同物种；目标组织是否纳入；参考标签是粗粒度还是细粒度；查询和参考是否可匹配基因；assay和归一化流程是否兼容；参考细胞数是否足以覆盖目标亚群。可把答案连同文档链接和责任人放入项目README。若关键条件不清楚，先做小规模试注释并标记探索性质，不把这次输出当最终细胞类型表。

针对冲突标签，抽取代表性细胞查看原始表达和标记基因，并比较各参考的置信信息。不要以“出现次数最多”作为唯一判定，因为多个参考可能共享相似训练来源，错误并不独立。保留不同参考对照结果后，由项目约定统一的命名和证据要求。最终表可包含accepted_label、supporting_reference、confidence_note与manual_review字段。

在交付报告中解释分块的影响

报告应说明分块大小、块数、分类顺序及训练对象是在何处建立的。若查询全量读入后再切块，应明确指出总体内存仍受全量RDS与所有参考影响；分块只针对后续分类数组。这能避免读者误把“分块”理解为任意规模都能低内存流式处理。

如果分块改变了剪枝策略或每块参考选择，结果变化不只来自计算分片。把各块原始输出与输入细胞清单一并保存，重新合并时用ID检查。对于计划重复使用的流程，可设计小的回归样本，升级SingleR、Bioconductor或参考后重新分类并对照标签变化，确保版本更新没有悄悄改变结论。

汇总报告中同时列出各参考覆盖到的细胞数、未定比例和发生冲突的标签，不只展示最后一列综合标签。该信息能让合作者判断结论是否依赖单一参考，并为日后更换参考提供可比较的基线。

如果不同参考的物种、组织或标签体系无法对应，应将各参考结果作为分开的注释证据呈现，先解决映射关系再形成统一标签。统一名称表本身也要版本化，避免不同分析批次使用了不同同义词规则。

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

SingleR官方书：多参考结果组合 · 来源资料

SingleR官方书：训练与分类拆分 · 来源资料

SingleR当前trainSingleR源码及索引边界 · 来源资料

SingleR当前classifySingleR接口 · 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。