AUCell批量打分教程：排序矩阵、基因集拆分与资源试验

完整操作教程 · 参考固定 · 往返恢复验收

固定一次细胞内基因排名，分批计算基因集AUC，保留匹配覆盖率和细胞顺序，分别测排名与打分峰值。

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。

AUCell批量打分可先把排名与基因集计算拆开。排名依赖表达矩阵和随机并列处理，计算完成后可复用；每次换一小组基因集都重新排名，会增加开销并可能引入并列顺序差异。本文固定同一细胞集合与基因全集，先生成排名，再按基因集分组计算，不把分组当作新的科学筛选。

明确输入和排名范围

输入或设置	需要记录	验收方式
expression.rds	基因乘细胞dgCMatrix；唯一名称	与目标细胞、基因范围一致
geneSets.rds	有唯一名称的字符向量列表	与矩阵基因ID相同；记录重叠率
aucMaxRank	本方案使用的排名截断	所有批次一致；检查检测基因分布
随机设置	种子及BiocParallel设置	排名只生成一次并保存

表达单位不是这里的唯一关键，基因全集、零表达和并列排名同样需要保留。AUCell会随机处理相同表达值，零值也可能排在尾部；不要把删去低表达基因后的分数直接与原全集比较。下例的百分之五仅为演示截断规则，需结合每细胞检测基因数量决定是否合适。

第一步只构建排名

```
library(AUCell)
library(BiocParallel)
z <- readRDS("expression.rds")
stopifnot(is(z,"dgCMatrix"),nrow(z)>0,ncol(z)>0,
 !is.null(rownames(z)),!is.null(colnames(z)),!anyNA(rownames(z)),
 !anyNA(colnames(z)),!anyDuplicated(rownames(z)),!anyDuplicated(colnames(z)),
 all(is.finite(z@x)),all(z@x>=0),
 !file.exists("ranking-output"),dir.create("ranking-output"))
detected <- Matrix::colSums(z>0)
write.csv(data.frame(cell=colnames(z),detected=detected),
 "ranking-output/detected.csv",row.names=FALSE)
set.seed(2026)
rk <- AUCell_buildRankings(z,plotStats=FALSE,
 BPPARAM=SerialParam(RNGseed=2026),keepZeroesAsNA=FALSE)
stopifnot(identical(colnames(rk),colnames(z)),identical(rownames(rk),rownames(z)))
saveRDS(rk,"ranking-output/rankings.rds")
writeLines(capture.output(sessionInfo()),"ranking-output/session.txt")
```

保存为rank.R。排名输出不应被当作稀疏输入的同等大小副本；官方实现可能构造完整排名矩阵。先记录排名对象和进程峰值，确认当前环境装得下，再进入打分。构建排名的旧nCores参数已不再控制并行，应使用对应版本支持的BPPARAM。

第二步分批打分并保留覆盖率

```
library(AUCell)
rk <- readRDS("ranking-output/rankings.rds")
gs <- readRDS("geneSets.rds")
stopifnot(is.list(gs),length(gs)>0,!is.null(names(gs)),
 !anyNA(names(gs)),all(nzchar(names(gs))),!anyDuplicated(names(gs)),
 all(vapply(gs,is.character,logical(1))),all(lengths(gs)>0),
 all(vapply(gs,function(x) !anyNA(x) && all(nzchar(x)) &&
!anyDuplicated(x),logical(1))),
 !file.exists("auc-output"),dir.create("auc-output"))
hit <- vapply(gs,function(x) sum(x %in% rownames(rk)),integer(1))
write.csv(data.frame(set=names(gs),total=lengths(gs),matched=hit),
 "auc-output/gene-coverage.csv",row.names=FALSE)
stopifnot(all(hit>0))
k <- ceiling(0.05*nrow(rk))
chunks <- split(seq_along(gs),ceiling(seq_along(gs)/100))
for (i in seq_along(chunks)) {
 idx <- chunks[[i]]
 a <- AUCell_calcAUC(gs[idx],rk,aucMaxRank=k,nCores=1)
 mat <- getAUC(a)
 stopifnot(setequal(rownames(mat),names(gs)[idx]),
   identical(colnames(mat),colnames(rk)),all(is.finite(mat)))
 saveRDS(a,sprintf("auc-output/chunk-%03d.rds",i))
 rm(a,mat)
}
saveRDS(list(seed=2026,aucMaxRank=k,chunks=chunks,set_names=names(gs)),
 "auc-output/settings.rds")
```

```
set -euo pipefail
/usr/bin/time -v -o rank.time.txt Rscript rank.R > rank.log 2>&1
/usr/bin/time -v -o auc.time.txt Rscript score.R > auc.log 2>&1
```

第二段存为score.R。100个基因集只是演示批次大小，不是内存承诺。全部排名仍常驻，分基因集只能限制本批AUC结果等部分对象，不能解决排名本身超内存。将来合并分数按基因集和细胞名称对齐，先查缺失及重复；如果最终完整分数矩阵很大，也可保持分块供下游读取。

验收重点是覆盖率和分布

官方教程将AUC用于观察基因集在每个细胞排名前端的富集。检查覆盖率、每细胞检测基因数、截断位置及分数分布，并解释工具关于缺失基因的警告。不能因为一个基因集有少量匹配就认定足够，也不能直接用统一AUC阈值给全部细胞贴类型。

排名失败先查矩阵范围、稠密化和实际RAM；打分失败先查基因集名称与交集。只有输入和科学范围固定后仍存在容量缺口，才评估更多内存。改aucMaxRank、删基因或换零值处理会改变分数，应另建方案并重新验证，而不是包装成无影响的资源调整。

保存可复核的批次索引

settings.rds中的chunks与set_names用于核对每个结果块覆盖哪些基因集，检查总数、重复和缺失。重跑某一块时读取同一排名文件与原截断参数，写新的尝试目录；不要在排名已重新生成后沿用旧块拼接，即使细胞名称看起来完全一样。

并将排名文件校验值随每块结果归档，避免无意拼接不同排名版本。

先核对表达矩阵和基因集输入

AUCell的输入表达矩阵需要确认行为基因、列为细胞或样本，并按该包函数接口准备。核对基因ID类型与基因集使用的命名系统一致；Ensembl ID、带版本号ID与gene symbol混用会导致交集过小。统计每个基因集与表达矩阵的交集数、重复基因和缺失比例，将清理规则固定后再算。不要在每个样本上临时改ID映射，否则评分不可直接比较。

rankings阶段的细胞数和参数

AUCell先在每个细胞内部对基因表达排名，再按基因集计算AUC。记录细胞数、基因数、矩阵稀疏性、表达层、随机种子、aucMaxRank与并行设置。默认rank阈值或AUC阈值并非对所有数据集通用；aucMaxRank应考虑每个细胞可检测基因数。若截断太低，低表达但有意义的基因可能无法贡献；太高则会纳入背景噪声。用参数敏感性分析说明阈值选择。

分步计算避免一次占满内存

先选择少量细胞和基因集跑小试验，记录rankings与AUC阶段的耗时和峰值内存。AUCell rankings对象可能接近矩阵规模，细胞数增加时开销会明显增长。对大对象按细胞分块时，确保每块都使用一致基因排序定义和参数，并用相同基因集计算；合并前检查列名、细胞顺序和AUC维度。并行worker复制对象也会提高RAM占用，应测试线程数而不是默认开满。

解释AUC和阈值时的边界

AUC是基因集成员在细胞表达排序前段的富集分数，不是绝对表达量，也不代表通路活性已被实验验证。AUCell_exploreThresholds可帮助查看分布并提出阈值，但多峰分布不一定存在生物学明确分界。可先按细胞类型画密度或分布图，再比较不同合理阈值下阳性比例。若无清晰阈值，保留连续分数并说明不强行二分。

细胞类型和样本构成会影响观察

细胞数量不均会影响总体分布和阳性比例。按样本、批次、细胞类型分层展示，避免将细胞当作独立生物学重复进行推断。评分差异应使用合适的样本层级统计方法，控制实验设计和混杂因素。AUCell分数高低不直接表示某个基因集的差异表达，也不应把一个细胞的阳性标签当作独立样本做显著性检验。

可复现报告与输出检查

保存基因集来源与版本、映射表、输入assay/layer、AUCell版本、随机种子、参数、矩阵散列和细胞元数据版本。输出检查包括：AUC列与细胞一一对应、基因集行名唯一、分数有限、分布图与阈值记录匹配。若更新基因集数据库或注释版本，应在新结果目录重新计算，并用固定的小样本对比变化。避免只保存阈值后的0/1矩阵而丢失连续评分。

准备多个基因集时的版本控制

每个基因集应有稳定名称、来源、版本、物种、ID类型和用途说明，并去除空ID及重复项。若同名基因集来自不同数据库版本，使用不同文件名和版本标记。计算每个集合与表达矩阵的交集比例，交集过低时先检查ID映射，不能通过改动AUC阈值补救。将映射失败的ID单独输出，便于后续更新注释。

阈值选择需要先看分布

为每个基因集画AUC密度或直方图，按细胞类型或样本分面检查是否存在明显双峰。若所有细胞的AUC集中在狭窄区间，强制阈值会产生不稳定的阳性标签。可比较自动阈值与研究者预先设定的阈值，列出阈值变化下阳性比例和主要细胞群是否稳定。AUCell探索阈值是辅助判断，最终采用规则应在分析方案中说明。

让并行计算可控

设置并行worker前先确认AUCell版本支持的后端和参数，再用小矩阵试跑一个worker和少量worker，对比耗时、RAM峰值、结果维度和分数差异。并行任务可能复制基因集、rankings或表达对象，RAM使用不一定按线程线性增长。保存随机种子和worker配置；若无法保证可重复排序，说明该实现的随机部分，并保持后续阈值和可视化基于同一轮结果。

常见错误一：AUCell分数全部接近零

先查看基因ID交集、矩阵是否取错assay/layer、表达值是否为适用于排序的数值，以及基因集是否为空。检查每个细胞有多少非零基因和排序截断范围；若大量基因名称不匹配，优先修ID映射。检查输入是否意外使用了标准化后残差或缩放值，而研究流程原本要求另一表达层。重新计算后用已知marker集合做合理性检查，但marker行为仍需结合细胞类型解释。

常见错误二：阈值阳性比例极端

确认基因集大小、交集数量、aucMaxRank和分布图范围。阈值探索结果可能因某个基因集分布连续或细胞群混合而不稳定，不能强行选分界。比较不同合理截断设置下连续分数和阳性比例；如果阳性只在某一个阈值突然出现，报告阈值敏感性并保留连续分数。不要通过删除低分细胞让阳性比例看起来符合预期。

对外报告的最小信息集

方法部分应写AUCell版本、表达矩阵来源与层、基因集数据库或自定义版本、基因ID映射、aucMaxRank、排序随机性设置、阈值选择方式和每个细胞的评分定义。图表注明分数是连续AUC还是阈值二值结果，并按生物学重复展示样本分布。可将完整参数和代码链接到项目归档，避免只写“按默认参数计算”导致无法复现。

从小试验扩大到全数据

先以少量细胞和少量基因集验证ID交集、评分输出和可视化；再增加到代表性的一个样本或一个细胞类型，测量rankings与AUC分别需要的时间和RAM。确认分数分布和对象维度合理后再处理全队列。并行度和内存阈值要结合样本数、每细胞检出基因数及worker复制开销确定。正式分析前保存输入与基因集校验值。

一个阈值不稳定的判读示例

假设一个基因集的AUC分布是单峰连续分布，探索工具却给出多个候选阈值。先分层看细胞类型与样本，检查候选阈值改变时阳性细胞比例是否剧烈变化。如果改变很大且缺少独立生物学依据，不应强行生成“活跃/不活跃”类别；保留连续分数并把分布展示出来。只有当阈值选择有稳定分布特征和研究预设支持时，才报告二值标签。

如何让分析能在另一台服务器复跑

将输入对象、基因集、ID映射表、代码、包版本和参数按项目相对路径保存。复跑时先检查对象行列名与校验值，再运行同样的ranking和AUC步骤。不同R/并行后端下结果可能有随机排序或浮点差异，使用已规定的比较摘要；若主要细胞群排名发生明显变化，回查输入层、随机种子、版本和worker设置。不要只提供最终热图作为复现材料。

不把评分直接转成通路活性

基因集AUC只描述其成员在细胞内排序前段的富集，不等于通路真实活性、蛋白功能或因果机制。高分或低分要与细胞类型、样本重复和独立实验结合解释。阈值二值化尤其容易掩盖连续差异，报告时保留分数定义和证据边界。

结果报告留存

保留连续AUC、阈值依据与分层图，读者才能复核最终标签并了解不确定性。

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

AUCell官方教程 · 来源资料

buildRankings参数与实现 · 来源资料

calcAUC参数与实现 · 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。