Milo邻域分析教程：图结构、邻域规模与稀疏计算排查

完整操作教程 · 参考固定 · 往返恢复验收

按构图、邻域和样本计数分阶段定位Milo内存问题，保留输入对应与验收记录。

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。

Milo内存不足时，先分清是构图、定义邻域、按样本计数，还是统计建模。表达矩阵稀疏，并不保证图、距离和后续对象始终很小。下面从已有PCA的SingleCellExperiment建立一份可检查的图与邻域计数，不直接跳到差异丰度结论。

先把细胞、样本与降维坐标对齐

准备已质控、标准化并完成PCA的sce-pca.rds。counts保留原始计数，sample_id代表真实生物学样本，不能把每个细胞当独立重复。PCA行顺序必须对应细胞列顺序；整合方式和保留维度先按研究设计确定，不在内存报错后随意换成另一种坐标。

阶段	检查内容	资源记录
输入	细胞ID唯一、稀疏计数、PCA与样本ID	细胞数、基因数、PCA维度
构图	k、维度、邻居算法与实际版本	墙钟、进程/作业RAM峰值
定义邻域	抽样比例、随机种子、邻域数量	对象大小与步骤耗时
样本计数	邻域×真实样本，样本没有丢失	矩阵维度、非零项与样本覆盖

分阶段运行并保留计数检查

将下列保存为milo_probe.R，在已验证的miloR环境执行。示例k=20、d=30、prop=0.1是一个明确的试跑协议，要求细胞数大于20且已有至少30个有效主成分，不是所有项目的最优设置。新目录避免覆盖旧对象，串行后端用于建立资源基线。

```
library(miloR)
library(SingleCellExperiment)
library(BiocParallel)
sce <- readRDS("sce-pca.rds")
stopifnot(is(sce, "SingleCellExperiment"), ncol(sce)>20,
          !is.null(colnames(sce)), !anyDuplicated(colnames(sce)))
cx <- counts(sce)
stopifnot(inherits(cx,"dgCMatrix"), all(is.finite(cx@x)),
          all(cx@x>=0), all(cx@x==floor(cx@x)))
pc <- reducedDim(sce,"PCA")
stopifnot(ncol(pc)>=30, identical(rownames(pc),colnames(sce)),
          all(is.finite(pc)))
meta <- as.data.frame(colData(sce))
stopifnot("sample_id" %in% names(meta),
          !anyNA(meta$sample_id), all(nzchar(as.character(meta$sample_id))))
stopifnot(dir.create("milo-probe-001"))
writeLines(capture.output(sessionInfo()),"milo-probe-001/session.txt")
record <- function(stage,x) {
  cat(stage,format(object.size(x),units="MiB"),"\n")
  x
}
set.seed(2026)
m <- record("graph",buildGraph(Milo(sce),k=20,d=30,
     reduced.dim="PCA",BPPARAM=SerialParam()))
m <- record("nhoods",makeNhoods(m,k=20,d=30,prop=0.1,
     refined=TRUE,reduced_dims="PCA"))
m <- record("counts",countCells(m,meta.data=meta,samples="sample_id"))
nc <- nhoodCounts(m)
stopifnot(nrow(nc)>0,ncol(nc)>0,
          setequal(colnames(nc),unique(as.character(meta$sample_id))))
if (as.double(nrow(nc))*ncol(nc) <= 1e6) {
```

```
  write.csv(as.data.frame(as.table(as.matrix(nc))),
            "milo-probe-001/nhood-sample-counts.csv",row.names=FALSE)
} else message("■■■■■■■■■■RDS■■■■■■■■")
saveRDS(m,"milo-probe-001/milo.rds")
```

上例只把较小的“邻域×样本”计数转成普通表；样本很多时应直接保存稀疏矩阵，先检查尺寸再决定是否导出。不要把细胞×邻域或基因×细胞大矩阵整体as.matrix。记录对象大小只能帮助比较步骤，不能代替进程总峰值；原对象、临时副本和包内部工作区仍可能同时存在。

```
set -euo pipefail
/usr/bin/time -v -o milo-probe-resources.txt \
  Rscript milo_probe.R > milo-probe.log 2>&1
```

内存和方法问题分别处理

构图阶段失败，核对坐标维度、邻居算法和并行设置；邻域步骤突然增大，记录实际邻域数、覆盖与重叠；计数异常先查sample_id及细胞对应。不要为省内存自行创建全细胞两两距离矩阵。降低k、维度或抽样比例会改变图与检验集合，需要方法敏感性检查，不能作为无影响的硬件优化。

按细胞任意拆批再拼邻域不等于联合构图，会改变跨批邻接关系。按细胞类型分析也需先有明确生物学问题并检查重复和样本覆盖；罕见群体不能只因数量小就删除。代表性子集试跑用于排错，不能直接外推完整队列的峰值或结果。

例如仅作容量演算：2万个邻域、12个样本的普通双精度计数主体约1.92MB；10万细胞×2万个邻域若被展开成同类稠密矩阵，主体约16GB，尚不含副本。相同的“转成矩阵”在不同槽位后果完全不同，应先查看dim、class和稀疏非零项，再决定导出方式。

进入统计检验前再验收一次

正式差异丰度还要核对设计矩阵、批次与条件是否混杂、每组生物学重复、低计数邻域及所用空间多重检验方案。不同miloR版本的距离与FDR接口有变化，应按安装版手册选择，不能照旧代码无条件生成全部距离。图和计数文件成功保存，仅说明这段计算完成，不说明实验设计已支持可靠的组间结论。

先确认设计单位是样本而不是细胞

Milo的差异丰度分析需要有真实生物学样本作为重复。sample_id应对应独立实验单位，而不是细胞条形码、批次内随机分组或技术重复的伪替代。分析前整理每个样本的条件、供者、批次和细胞数，确认设计矩阵可估计目标比较。若条件与批次完全混杂，邻域统计无法单靠增加细胞数解除混杂。

PCA、整合方式和保留维度会影响近邻图。记录降维生成方法、所用层或assay、纳入细胞集合和版本。若不同样本在上游单独变换后拼接，坐标尺度未必可比；如果使用整合空间，则确认其是否适用于当前差异丰度问题。不要在看到显著结果后反复更换整合设置。

图和邻域的结构检查

构图后检查每个细胞的邻居数、图是否有孤立节点、样本覆盖是否均衡，并查看不同k值下图连通性和邻域重叠变化。k和PCA维数是方法设定，不是为了省内存任意压低的旋钮。可进行预先定义的敏感性分析，在合理范围内比较主要邻域模式是否稳定，并完整报告参数。

makeNhoods生成邻域后，检查邻域数、每个样本进入各邻域的细胞数及未覆盖细胞。计数矩阵的行对应邻域、列对应样本时，要核对colData样本顺序；仅维度相同不足以证明映射正确。保存邻域对象、图参数和细胞ID散列，以便后续重建和追踪。

解释差异丰度结果

差异丰度检验基于样本层级的邻域计数及设计矩阵，不能把每个细胞当作独立重复进行简单检验。结果应结合样本数、每个邻域的计数、效应方向和多重检验校正解释。小样本或细胞数极不平衡时，统计功效和模型适配性会受限，即使软件返回p值也应谨慎。

将显著邻域映射回细胞类型、已知标记和样本来源，并检查是否由单个样本驱动。可做逐样本剔除或合理的设计敏感性分析，确认结论不完全依赖异常样本。若发现热点集中在低覆盖或边界邻域，回看邻域构建和样本覆盖，不直接把标签写成新的细胞类型发现。

资源测量与失败分流

分别记录PCA对象读取、buildGraph、makeNhoods、countCells和差异模型阶段的时间与峰值RSS。对象大小函数仅估算R对象，不包含R进程峰值和底层库工作区。多个邻域参数并行试跑会重复读入大对象；先串行建立基线，再按实测资源决定并发数量。

若构图阶段内存高，检查是否意外将稀疏counts展开或同时保留多个Milo对象；若countCells很慢，查看细胞ID和样本映射以及图规模；差异模型失败则先检查设计矩阵秩、样本数和对比设定。系统OOM、R错误和模型不可估计是不同问题，应从作业日志区分。

结果交付与复现

归档输入对象散列、PCA与图定义、k/d/prop、随机种子、样本设计、MiloR和Bioconductor版本、完整调用、资源日志和邻域计数表。保存差异丰度结果及其邻域映射，不只交一张UMAP图。下游整合或标签修订应另存版本，保留模型结果与人工注释的区别。

复核者应能从样本清单追到原始对象，并核对条件、重复单位和分析对比。报告中说明本流程未验证因果机制、不是差异表达检验，也不自动纠正批次混杂。若结果用于决策，需由熟悉实验设计的研究者复核统计单位和对比假设。

模型设计检查

差异丰度结果依赖样本设计矩阵。运行前检查设计矩阵是否满秩、目标对比是否明确、每组是否有足够真实重复；批次与处理条件完全重合时，无法从模型中单独估计处理效应。样本量小于模型需求或某组缺失时，先调整科学问题或补充实验设计，不通过增加线程解决统计不可估计。

不同样本细胞数差异可能影响邻域计数和检验稳定性。报告每个样本纳入细胞数、过滤规则和计数分布。敏感性分析可检查高低细胞数样本对结果的影响，但不得只删掉结果不理想的样本。对于离群样本，结合实验记录和质控指标说明是否排除及其影响。

版本和图缓存管理

Milo对象包含图、邻域及相关实验数据，跨版本保存后应核对对象结构和主要字段。若重建邻居图或更新miloR/BiocParallel，记录版本、图算法和参数并重新跑小样本基线。不要把旧版本图对象直接传给新版本模型而不做检查。

图缓存仅在细胞ID、降维和参数完全相同且散列匹配时可复用。样本增加、过滤变化或PCA更新都需要重新构图。把缓存条件写入文件名或manifest，避免多人共享目录里把不同研究输入的邻域图相互覆盖。

常见问题诊断

图邻域运行时间突然增加，先比较细胞数、降维维度、k值和构图后边数；同一参数但数据规模增加也会改变复杂度。若仅模型阶段报错，检查设计矩阵、真实样本数、目标对比及邻域×样本计数是否有全零列。保留完整错误和阶段时间，避免只截取最后一行。

邻域结果与已知细胞群不一致时，检查低维空间、样本混合、邻居算法和标签覆盖。Milo邻域可跨越传统聚类边界，不能按单个cluster名称替代邻域定义。将差异邻域映射回细胞时同时看样本组成和标记，避免把单一样本特异区域解释为通用状态。

资源评估说明

资源评估表记录细胞数、基因数、PCA维度、k、图边数、邻域数、并发、峰值RSS和各阶段耗时。申请额外算力时附上失败日志及作业内存额度，说明瓶颈是在构图、计数还是模型拟合；如果设计矩阵不可估计，增加资源并不能解决。

若设计矩阵中有供者配对或重复测量，应根据研究设计显式建立模型项；不要简单将各样本细胞数相加后当作独立样本量。缺失样本、重复标签错误或组别编码反了都会改变对比结果，运行前导出设计表并人工核对。

差异丰度报告建议列出邻域、log fold change、置信或检验信息、校正后显著性、各组样本计数及映射的细胞类型。可视化展示全局和局部结果，说明显著邻域并不自动等于一个新细胞类型。

Milo差异丰度结论只描述与输入细胞、建图方法和实验设计相符的统计关联。解释为细胞群扩增或减少时，要核对邻域内细胞注释、样本覆盖和效应大小；单一显著性阈值不应替代效应与不确定性的报告。

分析者应逐一核对细胞注释映射和样本计数矩阵，确认缺失标签没有被默认归为某个邻域。若标签来自人工修订，保存修订前后版本和依据，避免模型输入与结果展示使用了不同标签表。

如果需要在后续报告中引用某个邻域，保存稳定的neighborhood ID并注明其构建版本。邻域编号在不同运行间可能重新排列，不能单靠行号关联。可按代表细胞、邻域成员散列和相应差异结果建立索引，确保图表与统计表指向同一对象。

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

miloR构图接口 · 来源资料

miloR当前邻域实现 · 来源资料

miloR计数与检验手册 · 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。