比较全量与抽样步骤的资源消耗并规划运行环境

完整操作教程 · 参考固定 · 往返恢复验收

按assay、layer和计算范围规划大规模资源，分阶段测量sketch、学习与投影，同时核对结果含义。

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。

大规模单细胞分析采用磁盘后端后，选型问题会从“counts能不能装进内存”变成“哪些操作会读盘、哪些输出要留在内存、哪些结果需要全量”。资源规划应该跟随这张步骤图，而不能只按总细胞数给出固定规格。

本文比较两类工作范围：在全量数据上执行必要计算，以及先做sketch再投影。它们可能服务于不同目标，比较资源时必须同时记录结果含义。抽样流程更省资源时，也不能据此宣称与全量方法完全等价。

给步骤标注细胞范围和数据表示

先列出RNA完整数据、sketch细胞和目标亚群各自包含多少细胞。Seurat对象可以同时含有不同细胞集合的assay；不要只调用一次Cells或ncol就假定每个步骤都在相同数据范围运行。每次记录都应写assay、layer、矩阵类及实际维度。

```
library(Seurat)
obj <- readRDS("checked-object.rds")
rows <- list()
for (assay in Assays(obj)) {
  for (ly in Layers(obj[[assay]])) {
    m <- LayerData(obj, assay=assay, layer=ly)
    rows[[length(rows)+1L]] <- data.frame(
      assay=assay, layer=ly, features=nrow(m), cells=ncol(m),
      representation=paste(class(m), collapse="/"))
  }
}
write.csv(do.call(rbind, rows), "matrix-inventory.csv", row.names=FALSE)
writeLines(capture.output(sessionInfo()), "inventory-session.txt")
```

上述代码访问对象结构与矩阵维度，不主动稠密化。若对象引用外部矩阵，应先确认文件和版本可访问。缺失目录导致的读入失败与内存不足是不同问题，不能换大内存掩盖路径失效。

计算范围	可能保留的主要结果	需要实测的资源
完整counts存储与读取	磁盘矩阵和路径	容量、读取吞吐和缓存行为
sketch选择与预处理	抽取细胞及特征	构建过程峰值，不只最终对象
sketch PCA、图与聚类	低维与图结构	样本规模、维度和算法的占用
全部细胞投影	全量低维与标签	投影与邻居权重的峰值
亚群深入或差异分析	对应细胞与适用模型输入	可能再次物化或生成大对象

分别测三个资源阶段

将sketch准备、低维学习和全量投影保存成三个独立脚本或检查点。用同一版本、相同输入清单和明确参数运行，并记录退出状态、墙钟时间、任务内存峰值、目录最大占用和结果验收。对磁盘后端使用正确的保存与加载方式，不把中间RDS当作自包含文件。

```
mkdir -p logs
/usr/bin/time -v -o logs/sketch.time.txt   Rscript make_sketch.R >logs/sketch.out.txt
2>logs/sketch.err.txt
/usr/bin/time -v -o logs/learn.time.txt   Rscript learn_sketch.R >logs/learn.out.txt
2>logs/learn.err.txt
/usr/bin/time -v -o logs/project.time.txt   Rscript project_full.R >logs/
```

```
project.out.txt 2>logs/project.err.txt
```

三个脚本应包含已经验证的具体计算步骤，而不是仅改文件名。拆分的目的，是让资源记录对应实际范围。连续运行时还要检查磁盘缓存和其他负载；不能把第二次读取已缓存数据的速度全部归因为新的配置。

用输出维度理解内存数量级

若某个必要输出以双精度稠密形式保存，表达矩阵的元素载荷与特征乘细胞相关，低维表示则与细胞乘维度相关。下面只做载荷演算，不会分配大矩阵，也不是服务器最低配置；图、权重、原对象和临时结果需要另行测量。

```
payload_gib <- function(rows, cols) {
  as.double(rows) * as.double(cols) * 8 / 1024^3
}
# ■■■■■■■■■■■■■■■matrix-inventory■■■■■■
data.frame(
  output=c("■■■■■■■■■", "■■■■■■■"),
  GiB=c(payload_gib(20000, 300000), payload_gib(300000, 30))
)
```

资源比较要和结果验收一起提交

对于sketch方案，检查不同样本和已知细胞群的覆盖、重要标记、全量标签缺失及亚群的可信度。对于全量方案，明确哪些步骤真正需要所有细胞。需要样本级比较或差异表达时，应使用与统计设计相适配的数据，而不能把sketch细胞比例直接当作原始群体比例。

不要只比较单任务最小内存，还要考虑同一课题组的并发、数据库与工作盘、图像或对象保存、交接时的外部目录。一次投影成功也不能保证随后任意算法都支持磁盘矩阵；逐项验证比按细胞总数套固定套餐更可靠。

把尚未验证的步骤留在资源报告中

如果当前只完成sketch学习，还没有进行全量投影、亚群提取或正式统计，应在报告中逐项标注。后续新增一种算法时，重新确认它接收的数据层和矩阵类型，再安排试跑。这样既避免提前按最高资源长期占用，也不会把尚未测过的步骤藏在一个“整体已跑通”的结论里。

完成后形成一份分阶段表，列输入范围、表示方式、峰值、耗时、输出与未验证步骤。必要任务仍超出现有容量时，可以按这份表评估大内存CPU与存储方案。环境部署、后端目录交接和保存期限应写入约定范围，而不是由读者从“支持大数据”自行推断。

参考与版本核对

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

Seurat：BPCells磁盘矩阵与保存 · 来源资料

Seurat：sketch与全量投影工作流 · 来源资料

Seurat：ProjectData投影参考 · 来源资料

SeuratObject：保存与载入磁盘后端对象 · 来源资料

GNU Time：内存资源统计 · 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。