HiCExplorer资源估算：分辨率、稀疏矩阵与峰值内存

用hicInfo先查矩阵bin数和非零项，再在受控试跑中测HiCMatrix载入峰值；分清文件大小、稀疏存储和稠密内存估算。

Hi-C矩阵文件小，不代表载入或转换时只占同样大小的内存。评估HiCExplorer资源时，先看bin数、非零项和分辨率，再在代表性文件上测实际RSS。下面以HiCExplorer 3.7.2的cool或原生h5为例；请先确认命令和Python包来自同一环境。

先查矩阵信息，不要先转稠密

hicInfo可报告矩阵大小、bin数量、非零项及数值范围。cool格式默认可使用其元数据；h5格式需读取矩阵数据。若矩阵巨大，先确认当前版本的加载行为和节点余量。单独运行hicInfo也应记录峰值RSS，不能仅凭命令名称假定它不占内存。

指标	能回答什么	不能代替什么
磁盘文件大小	存储占用与传输量	不能代表解压后RAM
bin数量n	分辨率和基因组范围形成的矩阵维度	不能单独预测稀疏数据峰值
非零项nnz	稀疏矩阵已存储的接触数	不含对象、临时副本和算法工作区
峰值RSS	本次输入、版本和参数的实际进程内存	不保证其他分辨率和样本相同

先算理论边界，再测真实载入

若错误地把n×n矩阵变成float64稠密数组，仅数值本体约需8n²字节，不含额外对象；这只是风险提示，不是运行内存预测。稀疏CSR可根据data、indices、indptr估算已载入结构大小，但读取、格式转换和下游算法可能临时复制数据。因此报告CSR字节数时，应同时记录系统峰值RSS，不能把前者当作作业的内存申请值。

```
#!/usr/bin/env bash
set -euo pipefail
MATRIX=/data/rep1_10000.cool
OUT="$PWD/hic-resource-pilot-001"
mkdir "$OUT"
test -s "$MATRIX"
hicInfo --version > "$OUT/version.txt" 2>&1
/usr/bin/time -v -o "$OUT/hicInfo.time"   hicInfo -m "$MATRIX" > "$OUT/hicInfo.txt" 2> "$OUT/
hicInfo.log"
/usr/bin/time -v -o "$OUT/load.time"   python inspect_hic.py "$MATRIX" "$OUT/load"   > "$OUT/
load.log" 2>&1
```

inspect_hic.py可以加载一次HiCMatrix对象，确认其为方形稀疏矩阵后记录形状、nnz和三个CSR数组的字节数；不要调用toarray()。这一步本身会载入完整矩阵，必须在受控节点、代表性样本上做。若矩阵大到可能超过当前配额，就先用hicInfo和已知格式元数据估算，或申请有余量的短试跑任务。

为什么分辨率和格式会改变预算

固定基因组范围时，bin越小，bin数越多，矩阵规模迅速增加；实际RAM还与有效接触数、稀疏度、归一化和下游算法有关。不同分辨率的cool文件或mcool层级应分别核查，不能拿粗分辨率实测峰值直接代表最细层级。磁盘还要为输出、临时文件和备份留空间。

格式转换可能同时保留输入、输出和中间对象。避免把整矩阵转成dense；若只分析一个区域，优先使用支持区域提取的工具流程，但确认该子集是否满足下游算法对全局覆盖或归一化的要求。按染色体拆开计算也可能改变跨染色体互作和归一化结果，不能把拆分节省的内存当作等价分析。

做一个能复用的资源记录

记录HiCExplorer与HiCMatrix版本、矩阵格式、文件大小、bin数、nnz、分辨率、hicInfo和完整载入的峰值RSS、运行时间及临时盘占用。至少用最细计划分辨率和一个代表性样本试跑；如多样本并发，还要测实际并发时的总RSS和I/O。资源申请按观测峰值留余量，并在项目报告注明范围和输入条件，而不是给所有Hi-C矩阵套用同一RAM数。

参考：hicInfo官方文档；HiCExplorer 3.7.2矩阵格式说明；hicConvertFormat官方文档。

参考资料

hicInfo官方文档：矩阵大小、bin size与非零项 官方文档
HiCExplorer 3.7.2官方矩阵文件结构说明 官方文档
HiCExplorer 3.7.2官方格式转换参数 官方文档