核对合理过滤、分块策略和网络构建设置

确认WGCNA输入方向和质量，估算基因两两矩阵的量级，再用有记录的分块设置定位内存问题。

WGCNA 内存不足时，不能只按表达矩阵文件大小判断。相关性和拓扑重叠矩阵会随基因数快速增长，样本数、过滤后保留的基因数、相关方法及中间对象都会影响峰值。先明确矩阵维度和分析目标，再判断是否需要减少低信息基因、采用分块计算或增加作业内存；不要为了让程序跑完而随意改变软阈值或删去有生物学意义的基因。

先核对矩阵方向和过滤结果

WGCNA 的表达矩阵通常要求行为样本、列为基因。运行前记录样本数、基因数、数据类型、缺失值数量、归一化方法及筛选规则，并确认样本顺序与性状表一致。过滤低表达或低变异基因时，应依据预先确定的分析标准，并保存被过滤基因列表。输入方向弄反会让程序把样本当作网络节点，既改变科学问题，也可能造成完全不同的资源需求。

分块能控制计算规模，但会改变网络拆分方式

blockwiseModules 的 maxBlockSize 用于限制模块检测的分块规模；超过时，WGCNA 会先将基因预聚类到多个块。较小的块通常减少单块矩阵的内存压力，但会增加分块和结果管理工作；跨块关系也不等同于一次性对全部基因计算。因此应把分块设置、随机种子、软件版本和输入筛选规则一起记录。默认的 5000 是函数参数默认值，不是适用于所有研究的内存保证。

现象	先检查	调整方向
创建相关矩阵时 OOM	过滤后基因数、数据副本和并行进程	确认筛选逻辑，减少不必要副本并小规模试算
TOM 阶段内存或磁盘高	块大小、TOM 是否写盘、工作目录空间	降低块大小并监控存储峰值
运行结束但结果异常	样本方向、缺失值、性状样本顺序、警告	修正输入后重跑，不把异常模块解释成生物发现

在目标节点做可复现的试算

先用与正式数据同类型、同方向的表达矩阵子集估算资源；记录基因数、样本数、块大小、CPU、峰值内存、临时目录和运行阶段。试算适合比较计算规模，不用于选定最终模块结论。若分块前后模块数量或关键基因归属变化明显，应进一步评估稳定性和生物学解释，而不是只挑运行最快的一次。

```
library(WGCNA)
datExpr <- readRDS("datExpr.rds")
stopifnot(is.matrix(datExpr) || is.data.frame(datExpr))
cat("samples:", nrow(datExpr), "genes:", ncol(datExpr), "\n")
gsg <- goodSamplesGenes(datExpr, verbose = 3)
datExpr <- datExpr[gsg$goodSamples, gsg$goodGenes]
# softPower ■■■■■■■■■■■■■■■■■■■■■■
net <- blockwiseModules(datExpr, power = softPower,
  maxBlockSize = 5000, networkType = "signed",
  TOMType = "signed", saveTOMs = FALSE, verbose = 3)
saveRDS(net, "wgcna-network.rds")
```

示例假定 softPower 已依据研究设计和 WGCNA 前序诊断确定。正式结果还要检查样本和基因过滤报告、模块特征基因、模块与性状关联的样本数，以及关键模块在不同合理设置下是否稳定。保存完整脚本、sessionInfo()、输入矩阵哈希与运行日志，后续才能判断是生物变化还是参数变化造成的差异。

按证据决定资源，而非按矩阵文件大小猜配置

整理试算和正式运行的实际峰值，明确数据是否为稠密矩阵、是否保留 TOM、是否多进程并行，以及同一节点还有多少其他任务。再用这些证据估计资源余量。WGCNA 的分块设计降低大规模计算压力，但不意味着任何规模都能在个人电脑或固定内存配置下完成；当任务仍超限，应先确认输入与计算设计，再考虑更大内存节点或把计算提交到受控服务器。

参考资料

WGCNA 1.74 CRAN 官方参考手册 · 官方资料

WGCNA 官方 FAQ · 官方资料