宏基因组共组装还是分样本组装？先按研究目标判断

用样本关系、读数、菌株差异和下游比较目标判断组装策略，并在正式运行前估算线程、内存和临时空间。

多样本宏基因组项目做共组装，可能提高低丰度物种获得足够覆盖的机会；但把差异较大的样本全部合在一起，也可能让不同菌株的序列和丰度结构混在同一张组装图里。先问清楚研究目标：是为每个样本分别建立 contig，还是希望为某个生态组构建更完整的参考，再决定共组装范围。共组装不是自动省算力，也不是所有低丰度问题的通用解法。

先按样本关系划分候选策略

策略	适合考虑的情况	主要代价与风险
每个样本单独组装	关注样本间菌株差异或需要独立组装结果	多次重复计算，低覆盖样本可能难组装
同一生态组内共组装	样本背景相近，目标是为该组构建更完整的 contig	需解释组内混合对变异和丰度的影响
全体样本共组装	研究目标确需共享参考或序列目录	输入量和图复杂度可能明显增加，不适合替代样本级结果
先独立再合并 contig	希望保留各样本组装，再构建非冗余目录	还需去冗余、质量筛选和来源追踪

比较前先确定分析对象和分组

整理样本表，至少记录样本 ID、实验组、批次、FASTQ 路径、每端 reads 数和质控后数据量。技术重复是否合并、生物学重复是否保留独立，应由实验设计决定，不能为了让组装更完整而改变统计单位。若比较处理组与对照组，可先分别在组内试验共组装，再检查是否保留了各组特异 contig；是否需要跨组共享参考，要结合后续 mapping 与差异分析方案。

用小规模试跑估算资源和结果差异

从每个候选组中选代表性样本，用相同软件版本、预处理和资源设置分别比较单样本与组内共组装。记录总碱基数、耗时、峰值内存、临时盘峰值、contig 总长度和长度分布。试跑数据应采用固定规则抽取，并保留随机种子；若抽样量过小，不能代表复杂度或低丰度组分。共组装输入 reads 增加，并不意味着运行时间或内存会按样本数线性增加。

```
import csv
from pathlib import Path

rows = list(csv.DictReader(open("samples.tsv", encoding="utf-8"), delimiter="\t"))
assert rows and len({r["sample_id"] for r in rows}) == len(rows)
for row in rows:
    for col in ("r1", "r2"):
        p = Path(row[col])
        assert p.is_absolute() and p.is_file()
    assert row["group"]
# ■■■■■■■■ group ■■■■■■■■■■ MEGAHIT ■■■■■
```

示例只检查样本表和文件存在性，不会替你决定哪些样本可以共组装。正式任务可按 MEGAHIT 对 paired libraries 的输入格式为每组生成路径清单；参数以安装版本 megahit --help 为准。每个策略使用独立输出目录和唯一临时位置，保存完整命令与退出码。

用下游验收决定是否采用共组装

比较 contig 的长度和覆盖分布、目标基因或 marker 的恢复、分箱完整度与污染，并按样本回贴 reads 检查覆盖是否由少数高丰度样本主导。若共组装带来更长 contig，却丢失样本来源或把差异菌株合并，应评估这是否符合研究问题。最终归档组装策略、样本清单、软件版本、资源峰值和验收指标，让课题组能够复查为什么这样划分，而不是只留下一个结果目录。

参考资料

MEGAHIT 官方用法与多 library 输入示例 · 官方资料

MEGAHIT 官方内存设置说明 · 官方资料

SPAdes 官方 metaSPAdes 输入约束 · 官方资料