按数据特点比较组装路线与输出评价

先核对MEGAHIT与metaSPAdes的输入适用范围和内存参数单位，再以相同reads和预定评价比较结果。

做宏基因组组装时，MEGAHIT 和 metaSPAdes 都能处理短读长数据，但不能只按“哪个更快”选。MEGAHIT 面向大规模或复杂宏基因组设计；metaSPAdes 使用 SPAdes 的 metagenome 模式，官方要求其短读长输入为单个 paired-end library，并说明长读长混合组装仍属实验性流程。样本的复杂度、测序深度、菌株差异和研究目标，都会影响结果与资源。

先看数据和问题是否符合工具模式

确认是一个样本还是多个样本、单端还是双端、是否有长读长、是否预计存在较高菌株差异。不要把多个生物学样本直接合并成一个 library 来省算力，除非研究设计明确要求合并；这样会改变后续丰度解释。检查 reads 质控、接头残留和配对关系，保存输入校验值与过滤记录。

判断点	MEGAHIT	metaSPAdes
使用方式	为宏基因组及大规模数据设计；支持多个 paired 与单端库组合	SPAdes 的 meta 模式；短读长仅支持一个 paired-end library
资源设置	按安装版本查看 -m 含义和可用内存模式	显式设定 -t 与 -m，确保不超过调度器申请
复杂度差异	可试验官方提供的复杂宏基因组 preset	--frugal 可能降低内存，但会影响 contiguity

先做同一输入的小规模对照

如果要比较工具，使用同一份代表性 reads、同一预处理、相同的节点资源和存储位置，分别创建新输出目录。记录软件与数据库版本、CPU、峰值内存、临时盘峰值、耗时和退出码。小规模输入只能验证流程和大致瓶颈，无法替代全数据组装，也不能用一项 contig N50 指标决定哪个结果更好。

```
set -euo pipefail
: "${R1:?set forward reads}"
: "${R2:?set reverse reads}"
: "${OUT:?set a new output directory}"
: "${THREADS:?set allocated thread count}"
: "${MEM_GB:?set memory cap within allocated RAM}"
test -r "$R1" && test -r "$R2"
test ! -e "$OUT"
spades.py --meta -1 "$R1" -2 "$R2" \
  -t "$THREADS" -m "$MEM_GB" -o "$OUT"
```

命令只展示符合 metaSPAdes 单个双端短读长库的调用方式；MEGAHIT 参数名及内存选项应以安装版本的 megahit --help 为准。若数据是多库组合，不要照搬此 SPAdes 示例。

用组装质量和下游目标做决定

对照组装的总长度、contig 数和长度分布、覆盖情况、预期基因或 BUSCO/marker 指标，并按课题需求观察分箱、分类或功能注释是否稳定。不同 assembler 产生的 contig 不能只按长度排名；较高连续性也不自动代表分类更准确。若比较结果差异明显，先检查输入过滤、菌株多样性和参数版本，再判断差异来自算法还是数据处理。

最后将选择依据写进记录：数据特征、运行模式、参数、实际资源峰值、组装验收和适用限制。MEGAHIT 的内存设置说明指出其图构建阶段会受可用内存和扫描次数影响；该文档较早，务必对照安装版帮助。配置需求应由目标数据试跑测得，不能从工具名称推断固定内存。

比较时把质量指标放回研究问题

若目标是物种分箱，除了组装连续性，还应观察分箱数量、完整度和污染；若目标是发现功能基因，则检查目标基因或通路是否得到可解释的序列支持。不同指标有不同偏好，不宜把一项指标作为唯一胜负标准。最好先设定评价标准，再查看两个工具的输出，避免事后挑选最有利的指标。还要保留原始 reads、过滤后 reads 和样本元数据，确保组装能够回溯到正确样本。

如果是高复杂度环境样本，可以使用官方支持的模式做小规模参数试跑，但应记录软件版本和全部命令；高菌株多样性会使图结构与组装难度发生变化，不能用低复杂度样本的结果推算。工具比较的结论只适用于当次输入、数据库与设置，换样本后需重新核对。

参考资料

MEGAHIT 官方项目与使用说明 · 官方资料

MEGAHIT 内存设置官方 Wiki · 官方资料

SPAdes 官方运行参数说明 · 官方资料