OrthoFinder多物种资源试跑：蛋白质集检查与阶段预算

完整操作教程 · 参考固定 · 往返恢复验收

OrthoFinder资源需求既取决于物种和蛋白质数量，也取决于后续比对、基因树与物种树路线。本篇选择OrthoFinder 3的完整MSA分析，先清点蛋白集，再固定一次可复查试跑。核心物种增量分配另有流程，不能把更省资源的另一种算法当作本次全量分析的等价结果。

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。

OrthoFinder资源需求既取决于物种和蛋白质数量，也取决于后续比对、基因树与物种树路线。本篇选择OrthoFinder 3的完整MSA分析，先清点蛋白集，再固定一次可复查试跑。核心物种增量分配另有流程，不能把更省资源的另一种算法当作本次全量分析的等价结果。

每个物种一套有来源的蛋白质

将至少三个物种的蛋白质FASTA放入proteomes目录，统一以.fa结尾。通常按研究设计选择每个基因的代表转录本，并记录选择方法；不能把多个转录本都当成独立基因后直接解释拷贝扩张。每个文件内ID唯一，跨物种同名ID的身份需连同物种保存，后续整合时不要仅按短ID连接。

记录项	为什么需要	后续核对
物种、注释版本及蛋白集SHA	冻结分析对象	与日志物种清单一致
序列数和总氨基酸数	暴露异常注释或输入膨胀	不只数FASTA文件
-M / -S / -A / -T	固定推断路线和外部工具	不要只记OrthoFinder版本
-t / -a	区分外部与内部并行	多任务时预算叠加

先输出一个可以审阅的输入清单

以下检查器使用Python3.11以上，按行读取，拒绝重复ID、空序列、空文件和在首个标题前出现的序列。它检查基本结构，不负责验证蛋白质预测质量；异常字符、过多短片段与污染需另行评估。运行时要求inventory尚不存在，避免把新旧输入清单混在一起。

```
import hashlib,json
from pathlib import Path
rows=[]
files=sorted(Path("proteomes").glob("*.fa"))
assert len(files)>=3
for path in files:
    seen=set();length=0;total=0;name=None
    with path.open() as f:
        for raw in f:
            line=raw.strip()
            if not line:continue
            if line.startswith(">"):
                if name is not None:assert length>0
                fields=line[1:].split();assert fields
                name=fields[0];assert name not in seen
                seen.add(name);length=0
            else:
                assert name is not None
                assert line.isascii() and line.isalpha()
                length+=len(line);total+=len(line)
    assert seen and length>0
    with path.open("rb") as f:digest=hashlib.file_digest(f,"sha256").hexdigest()
    rows.append(dict(species=path.stem,file=str(path),proteins=len(seen),residues=
total,sha256=digest))
with open("proteome-inventory.json","x") as f:json.dump(rows,f,indent=2)
```

```
print("species",len(rows),"proteins",sum(r["proteins"] for r in rows))
```

固定路线完成一次全流程试跑

已安装环境须通过官方测试，并具有DIAMOND、MAFFT、FastTree及该版本其他依赖。以下两线程和一个内部并行是教学起点，不是推荐配置。输出路径必须全新；保存帮助输出与安装版本，不能从网上某个版本的默认值推断本机行为。

```
set -euo pipefail
test -s proteome-inventory.json
test ! -e orthofinder-baseline-001
test ! -L orthofinder-baseline-001
orthofinder -h > orthofinder.help.txt
/usr/bin/time -v -o orthofinder.time orthofinder \
  -f "$PWD/proteomes" -M msa -S diamond -A mafft -T fasttree \
  -t 2 -a 1 -o "$PWD/orthofinder-baseline-001" > orthofinder.console.txt 2>&1
```

按实际结果目录验收阶段

从日志取得本次Results目录，不硬编码日期。检查实际物种与输入清单一致，读取统计、基因树和物种树相关结果，确认流程不是停在搜索结束。当前官方优先建议使用层级直系同源群HOG；N0.tsv对应根节点，其他节点文件有不同范围。旧Orthogroups.tsv已被标记为弃用结果，不能不看版本直接沿用旧教程。

记录搜索、MSA/基因树和汇总阶段的起止时间、峰值内存与工作盘。-t主要控制搜索、比对和建树等外部任务，-a控制内部分析并行；同时提高两者可能增加高内存任务竞争。单个外部程序的峰值不能直接代表整个任务树的峰值，应结合调度器或cgroup数据。

先挑具有代表性的物种集合完成整条路线，再评估全量预算；只运行几分钟搜索不能推算后面树推断的耗时。出现某个物种分组异常时先检查蛋白集与代表转录本方案，树推断失败则定位对应家族及其比对。扩容或增加并发之前，保留一个完整成功任务的版本、参数、日志和验收表，后续比较才有依据。

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

OrthoFinder官方路线与输出 · 来源资料

OrthoFinder命令解析与并行设置 · 来源资料

Python file_digest版本与文件哈希 · 来源资料

OrthoFinder 官方使用教程与结果说明 · 来源资料

OrthoFinder 官方结果目录解释 · 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。