核对GTDB-Tk与参考库版本并检查内存日志

区分pplacer内存压力、参考库不兼容和其他终止原因，先完成版本与数据检查，再尝试有记录的线程和scratch策略。

GTDB-Tk 报 reference data 版本不兼容、分类运行到 pplacer 失败时，先核对软件包与数据库是否配套，不要先重装整个环境。当前 GTDB-Tk 2.8.0 安装页说明需要外部 reference data，并列出了 GTDB release 的兼容版本范围；分类命令还依赖 Prodigal、HMMER、pplacer、skani 等程序。环境变量、数据库路径或 PATH 错误，可能表现为运行失败，但增加内存并不能解决。

确认版本和参考数据库路径

在实际提交作业使用的 conda、容器或模块环境中运行检查。记录 GTDB-Tk 版本、GTDB release、数据库完整路径和安装来源，并确认作业环境与交互终端使用的是同一套变量。GTDB-Tk 官方说明要求设置 GTDBTK_DATA_PATH 指向解压后的参考数据目录；不同 release 并非都能被当前软件版本使用。

```
set -euo pipefail
command -v gtdbtk
gtdbtk --version
printf 'GTDBTK_DATA_PATH=%s\n' "${GTDBTK_DATA_PATH:-unset}"
test -n "${GTDBTK_DATA_PATH:-}"
test -d "$GTDBTK_DATA_PATH"
gtdbtk check_install
```

用官方检查命令区分缺依赖与数据库损坏

检查结果	优先排查	处理后再确认
命令找不到	当前作业是否激活正确环境、PATH 是否一致	命令路径、软件版本与依赖版本
reference data 路径不存在	环境变量是否指向解压目录而非压缩包	数据库路径和权限
HASH MISMATCH	数据库下载、解压或复制是否不完整	按官方 release 校验值重新核对
pplacer 内存失败	阶段、物理内存、pplacer 线程和树模式	资源设置与工作目录空间

gtdbtk check_install 可验证第三方软件是否在 PATH 中，并检查参考数据文件完整性。它通过不能证明输入基因组质量符合分类要求，也不能代替作业期间的资源监控。

分类前整理输入和运行记录

官方 classify_wf 接受 FASTA 基因组目录或 batchfile。目录模式默认处理 fna 扩展名；压缩文件需要明确设置扩展名。检查每个文件是否为核酸 FASTA，文件名是否能作为稳定 ID，并保存样本清单与校验值。输入基因组完整度、污染和 marker gene 缺失会影响结果，分析前应结合质量控制工具评估。

正式运行时单独指定新输出目录，记录命令、软件与数据库版本、输入清单、日志、退出码和作业资源。若失败发生在 pplacer 阶段，应单独查看该步骤的错误和内存指标；reference 数据损坏、工具缺失与 RAM 不足是不同问题，需分别处理。复现归档中保留所用 release 编号，不能只写“最新版数据库”。

注意数据库与软件的兼容范围

GTDB-Tk 的 reference data 通常占据大量磁盘，安装页给出的当前参考包规模约 100 GB；下载压缩包、解压副本和运行输出可能同时存在，磁盘规划时要分开估计。不要在共享环境中随意覆盖公共数据目录，也不要因其他 release 的环境能运行就推断当前环境兼容。保留 release 名、来源地址、校验信息和安装时间，可以让同一分析在另一台机器上复现。

如果 check_install 显示数据库文件哈希不匹配，先检查下载是否中断、解压是否完整以及当前 GTDBTK_DATA_PATH 是否指向预期 release。不要把不同 release 的文件手动混在一个目录里“补齐”；按兼容列表重新准备完整版本后再次运行检查。

把检查结果纳入项目归档

项目记录中保存 check_install 完整日志，并注明该日志对应的 GTDB-Tk 环境和 reference 目录。如果服务器由多人共用，应记录数据库目录的只读权限和维护人，避免单个作业更新 release 后影响其他课题。每次切换软件环境后重新执行依赖检查，确保批处理作业没有加载到另一套旧版本。

参考资料

GTDB-Tk 2.8.0 官方安装与硬件要求 · 官方资料

GTDB-Tk classify_wf 官方文档 · 官方资料

GTDB-Tk 官方 FAQ：pplacer 内存问题 · 官方资料

GTDB-Tk check_install 官方文档 · 官方资料