用合法可用环境试跑并确定GPU和存储配置

完整操作教程 · 参考固定 · 往返恢复验收

以已获允许使用的AlphaFold3环境为例，分开执行CPU数据搜索与GPU推理，保存带MSA输入及输出验收记录。

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。

AlphaFold资源试跑应先明确允许使用的软件与参数，再把数据准备和模型推理分开测量。本文以已经安装兼容AlphaFold3环境、已取得符合用途条件的模型参数和完整数据库为前提。它不提供参数下载替代途径，也不把可运行环境等同于模型有效性。

先固定一份原生输入

准备经过官方输入规则检查的fold_input.json，记录所有链、拷贝、分子类型、修饰、配体和随机种子。软件提交、镜像、数据库发布与模型参数来源一起留档。资源比较时保持同一任务，不通过删链或减少实体把任务改小后宣称原问题解决。

官方流程支持分别控制数据搜索与推理。常规数据管线执行遗传与模板搜索，可在合适CPU环境完成；模型推理需要兼容GPU。先在当前版本帮助中确认两个开关，不把AlphaFold2的命令直接搬到这个入口。

先运行数据管线

下面命令在准备好的AlphaFold3执行环境中运行，AF3_ROOT为源码目录，数据库与模型目录均已正确挂载。DATA_OUT必须是新的输出位置；GNU time记录这一段的进程资源。环境安装和数据库下载不混入本次计时。

```
set -eu
: "${AF3_ROOT:?■■■■■AF3■■■■}"
: "${INPUT_JSON:?■■■■■■■■JSON}"
: "${MODEL_DIR:?■■■■■■■■■■■■}"
: "${DB_DIR:?■■■■■■■■■■■■}"
: "${DATA_OUT:?■■■■■■■■■■■■■}"
test -r "$INPUT_JSON" && test ! -e "$DATA_OUT" || exit 2
/usr/bin/time -v -o "$DATA_OUT.time.txt" \
  python "$AF3_ROOT/run_alphafold.py" --json_path="$INPUT_JSON" \
  --model_dir="$MODEL_DIR" --db_dir="$DB_DIR" --output_dir="$DATA_OUT" \
  --run_data_pipeline=true --run_inference=false \
  > "$DATA_OUT.stdout.log" 2> "$DATA_OUT.stderr.log"
```

检查完成状态和日志，找到本任务带MSA及模板数据的_data.json。以实际输出文件为准，不按猜测的任务名拼路径；多个任务时逐一关联。若使用外部MSA或模板路径，也需保留对应文件与来源，迁移到另一机器后路径必须仍可解析。

用准备结果单独运行推理

DATA_JSON指向刚验收的数据阶段JSON，INFER_OUT为新的输出位置。以下仍使用同一源码与模型环境，并在获配GPU可见的会话内执行；不改变输入实体或方法参数。

```
set -eu
: "${AF3_ROOT:?■■■■AF3■■■■}"
: "${MODEL_DIR:?■■■■■■■■■■■■■■■}"
: "${DATA_JSON:?■■■■■■■■■■■JSON}"
: "${INFER_OUT:?■■■■■■■■■■■}"
test -r "$DATA_JSON" && test ! -e "$INFER_OUT" || exit 2
/usr/bin/time -v -o "$INFER_OUT.time.txt" \
  python "$AF3_ROOT/run_alphafold.py" --json_path="$DATA_JSON" \
  --model_dir="$MODEL_DIR" --output_dir="$INFER_OUT" \
  --run_data_pipeline=false --run_inference=true \
  > "$INFER_OUT.stdout.log" 2> "$INFER_OUT.stderr.log"
```

阶段	观察资源	验收材料
数据搜索	CPU、RAM、数据库读取和工作盘	日志、MSA/模板及对应JSON
编译与推理	GPU、显存、主机RAM和缓存	模型日志、结构和置信度文件

阶段	观察资源	验收材料
完整交付	两阶段时间、传输及结果留存	输入版本与全部任务状态

若搜索成功而推理失败，保留已验收的数据JSON，从同一关口重试；若数据库版本或输入改变，则重新生成数据，不能无说明地沿用旧MSA。

分别验收容量和结构结果

第一次推理可能包含编译，后续复用条件与版本有关。比较设备时写清是否包含编译、缓存与数据搜索，不能把仅推理时间当全项目耗时。GPU监控采样可能遗漏短峰，记录设备、采样间隔及同卡负载，不将一次截图当容量保证。

结果应包含预期链与实体、可读结构、相应置信度和排名记录。pLDDT、PAE等反映模型置信信息，不是实验确认；复合物还要检查界面、冲突及研究相关证据，不能由排名分数推断结合活性。

先验小型环境输入，再以正式代表规模和计划并发复验。将搜索与推理各自需求、数据库和结果保留空间写清，确认实际缺口后才确定资源；新的分子组成、参考或模型版本应重新核对。

把数据库准备和模型推理分开规划

AF3 的完整本地流程包括输入准备、遗传数据库搜索或 MSA/模板准备、特征生成、模型推理与结果写出。数据库下载、解压和索引会占用大量持久化存储与网络时间；遗传搜索阶段可能需要较多 CPU 和主机 RAM；模型推理则更依赖兼容 GPU、显存和 CUDA/JAX 环境。不要把整套 AF3 的资源需求简单归结为一张 GPU 规格表，也不要用一次 GPU 推理时间估算包含数据库准备的全流程工期。

官方安装文档提示完整遗传数据库最高可能需要约 1 TB 磁盘空间，SSD 更合适；较长目标的遗传搜索阶段建议至少 64 GB RAM，并给出了 A100 80 GB 与 H100 80 GB 对特定输入规模的验证信息。这些是官方测试条件，不是所有复合物、输入长度与数据库组合的通用保证。先列出计划使用的数据库版本、权重、分子类型、最长 token 数、并发预测数、数据保留期限和可用 GPU 型号，再用真实项目的代表输入试跑。

下载、解压与升级要有版本边界

数据库文件可能同时有压缩包、解压目录、索引和运行缓存，下载过程会暂时保留多份副本。规划磁盘时不要只按压缩下载大小计算；还要纳入临时空间、用户配额、日志、输入输出结构和数据库升级副本。数据库放在可靠、吞吐稳定且长期可用的存储上；软链接或挂载路径要在实际计算节点验证，不能只在登录节点上检查。

下载失败时记录文件大小、校验和与失败阶段，不要继续使用不完整数据库。更新数据库或权重时使用新的版本目录，校验后再切换配置，避免原位覆盖正在供多个任务使用的数据库。AF3 权重和数据库有各自的使用条款，下载成功不代表可以用于任意商业或再分发场景，项目开展前应核对官方许可。

推理需求随输入而变

显存需求会随输入 token 数、蛋白长度、链数、配体或核酸类型及推理设置改变。先用短输入验证环境，再用接近目标长度和复杂度的输入测量峰值显存、主机 RAM、耗时和失败率。只测一个短蛋白不能证明长蛋白或多链复合物能运行。多个并发任务会占用额外显存和主机资源，应从低并发逐渐增加并记录峰值。

输入严格遵循 AF3 官方 JSON schema。保存输入、序列或配体来源、模型 seed、数据库版本、权重版本和代码版本。输入格式错误、残基标识不兼容与资源不足会表现为不同失败；要根据日志定位，而不是一律增加 GPU。预测结果目录要按样本和 seed 隔离，防止同名文件覆盖。

预测结构仍需要生物学验证

AF3 输出的结构、置信度和误差估计属于模型预测。pLDDT、PAE 等量可帮助判断模型对局部结构或相对位置的信心，但不能代替实验结构、功能测定或生化验证。复合物还要检查界面置信度、链间位置、已知结构域边界和配体几何；低置信区域不应仅凭可视化作强结论。比较不同 seed 时预先定义指标，保留全部运行记录，而不是只展示最符合预期的结果。

报告模型和数据库版本、输入构建方式、seed、硬件和后处理方法。若结构用于实验设计，应与保守性、已有实验数据和其他独立证据结合，并指出不确定区域。预测结果不意味着实验验证，许可条款、数据库来源和访问控制也应纳入交接。

建立可复用的资源台账

每次运行记录阶段耗时、CPU 与 RAM 峰值、GPU 型号及显存峰值、数据库读取时间、输出大小、失败原因和输入 token 数。分开统计数据库准备、搜索、推理和后处理，才能判断瓶颈来自网络、磁盘、CPU/RAM、GPU 还是输入构造。若使用预先生成的 MSA 或模板，记录来源、生成软件和对应输入校验和。

多个课题组共享环境时，按项目隔离输入与结果权限，明确数据库只读共享、GPU 并发、数据保留与清理责任。原始序列是否允许放到租赁服务器，应按单位数据管理和合作协议确认。配置以代表性真实输入实测为依据，不凭 AF3 这个软件名称猜规格。

分别验收下载、特征与推理阶段

用阶段状态表避免混淆：数据库下载校验完成、数据库可读取、MSA/模板阶段结束、特征生成结束、GPU 推理完成、输出结构可解析。对失败任务记录阶段、错误日志和已占用空间，重跑前确认是否可以复用前序结果。官方示例路径只作为参考，应在目标服务器真实目录和计算节点上逐项验证权限、挂载与读写速度。

如果课题组只需要少量结构预测，可比较官方在线服务与本地部署的输入限制、隐私要求、许可和等待时间；如果需要批量、高可控运行，再根据样本长度、并发与数据库保留要求规划本地环境。不要把平台选择简化成单卡价格比较，数据库维护和合规成本也属于总成本。

预测结果目录要与原始输入对应

一个结构文件至少应能追溯到原始 query、输入 JSON、数据库版本、model seed 和输出日志。可以给每个任务生成短而稳定的任务 ID，并在 CSV 或 JSON 清单中保存完整映射，避免在文件名里直接放敏感样本信息。一个 seed 输出多个候选结构时，保留候选编号和选择规则，不要只把所有结果重命名为 final.pdb。

批量后统计成功数、失败数、输入数、输出结构数和可解析结构数，差异应逐条说明。服务器运行中断时先查看阶段标记和最后日志，再决定是否从已完成的数据库搜索或特征阶段继续；不要仅看输出目录里有文件便断言流程已完成。归档前执行只读清单和校验，避免在交付前无意覆盖旧结构。

评估数据库共用的维护成本

团队共用数据库可以减少重复下载，但需要确定谁负责更新、谁有写入权限、任务如何避免读取正在更新的目录，以及版本升级后旧项目能否继续复现。将正式数据库设为只读，更新到新目录后完成校验，再由管理员切换软链接或配置。不要让多个用户各自手工改同一份数据库路径。

维护台账记录版本、更新时间、占用空间、校验状态和许可来源。若数据库更新影响结果，应能将运行映射回当时使用的版本。数据库下载与模型权重的许可并不相同，清理、迁移和共享都需遵守其各自要求。

按目标复合物验证软件链路

对于含蛋白、核酸或配体的输入，分别检查每类分子是否符合当前 AF3 输入格式，配体化学标识和原子信息能否被识别。先在不含敏感信息的代表样例上运行，从输入 JSON 到最终结构逐阶段核对；如果某类复合物失败，不要把简单单体的运行成功推广为整套工作流已验收。对正式样本保存每种分子链的对应关系、输入序列摘要和输出链映射。

为下一轮计算建立容量依据

将数据库所在盘的读取速度、每个输入的 token 数、预测 seed 数、GPU 显存峰值和失败重跑次数纳入容量估算。按短、中、长输入分层，而不是用平均序列长度估全部任务。若某类多链输入失败率较高，先检查输入格式和显存，再决定增加设备；若数据库读取占主要时间，应先评估存储吞吐。记录每次估算的条件和观察日期，软件或数据库升级后重新测量。

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

AlphaFold2官方安装与数据库配置 · 来源资料

AlphaFold3官方运行与分阶段执行 · 来源资料

AlphaFold3输入格式 · 来源资料

AlphaFold3模型参数使用条款 · 来源资料

AlphaFold3输出文件与置信度说明 · 来源资料

AlphaFold 3 官方安装、数据库与硬件说明 · 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。