整理序列批处理、MSA来源与结果检查流程

完整操作教程 · 参考固定 · 往返恢复验收

从单体FASTA的唯一标识与字符检查开始，明确公共MSA服务的使用，再逐项验收结构和置信度结果。

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。

ColabFold批处理首先要让每条输入与结果稳定对应，再考虑运行多少个任务。这里以普通蛋白单体FASTA为例，依次检查序列清单、运行环境、MSA来源和结果。复合物、非蛋白分子或其他模型入口应按对应格式另行准备，不能仅修改文件名套用。

让序列名称成为可追溯的任务标识

每条序列保留来源、版本和研究含义，使用唯一且简单的ID，不把长注释直接当路径。序列末尾终止符、比对间隙或异常字符要先确认原因；不能为了通过检查随意删除残基，删除后已是另一个输入。

以下标准库脚本限定二十种标准氨基酸字符的非空单体序列，不处理带冒号的多链格式。保存为check_queries.py，执行python check_queries.py queries.fasta，输出仅有ID与长度，不打印完整序列。

```
import re, sys
records = []
name = None
parts = []
with open(sys.argv[1], encoding="utf-8") as handle:
    for line in handle:
        line = line.strip()
        if not line:
            continue
        if line.startswith(">"):
            if name is not None:
                records.append((name, "".join(parts)))
            name, parts = line[1:].split()[0], []
        else:
            if name is None:
                raise ValueError("sequence before header")
            parts.append(line.upper())
if name is not None:
    records.append((name, "".join(parts)))
assert records
ids = [x[0] for x in records]
assert len(ids) == len(set(ids))
allowed = set("ACDEFGHIKLMNPQRSTVWY")
for name, sequence in records:
    assert re.fullmatch(r"[A-Za-z0-9_-]+", name)
    assert sequence and set(sequence) <= allowed
    print(name, len(sequence))
print("queries", len(records))
```

若实际序列含模糊或非标准残基，本脚本会停止，应按所用模型的输入规则判断，而不是自动替换成某个氨基酸。检查通过只证明这些基本格式条件，不代表序列生物学正确，也不说明适合所选结构模型。

先跑一个小批次

在已经安装兼容ColabFold、JAX和所需模型环境的会话执行，结果目录应尚不存在。下面命令给FASTA输入时，默认会向公共MSA服务查询序列，再在本地推理；“本地运行”不代表全程离线。

```
set -eu
python -c "import jax; from importlib.metadata import version;
print(version('colabfold')); print(jax.__version__); print(jax.devices())"
colabfold_batch --help > colabfold-help.txt
test ! -e predictions-pilot
colabfold_batch queries.fasta predictions-pilot \
  > predictions-pilot.stdout.log 2> predictions-pilot.stderr.log
```

仅对允许发送到该服务的序列使用此路线，并遵守服务使用条件。需要离线或较大规模搜索时，改用已经配置好的本地MSA数据库流程，不通过大量并发请求规避服务限制。模型设置、随机种子和数据库来源都写入本轮清单。

检查层面	应核对	失败时先处理
输入清单	唯一ID、序列版本与条数	重名、空序列和格式
MSA	来源、查询完成状态与对应ID	网络、服务状态或本地数据库
推理	实际设备、模型日志和退出状态	框架兼容、显存或输入规模
结果	结构、置信度、排名及任务覆盖	缺项、旧文件或未解释的异常

建议建立三列状态表：输入ID、实际结果路径、验收结论。每个输入必须有一行，尚未完成的也保留并写明原因；结果目录同时保存原始FASTA和运行设置。这样重跑时能只选择明确的失败任务，避免把旧模型混进新批次，或把同一输入的多个模型误当成多个成功样本。

按输入逐项验收结果

核对每个任务是否完成，并将结构文件、评分JSON和日志关联回原ID；输出命名以安装版本为准。不要仅统计PDB数量，因为一个输入可能产生多个模型。缺失项与失败原因单独登记，重跑使用清楚的新批次。

检查链长、残基对应、低置信区域及模型间差异。pLDDT、PAE和模型排名不是实验结构确认，更不能直接当作活性或结合能力。先确认这批输入与输出可追溯，再按长度、MSA规模和模型设置分组扩展，避免把最短序列跑通当成整批容量已验证。

把 MSA 搜索与结构推理当作两个任务

ColabFold 可以在线搜索 MSA，也可以使用本地数据库和搜索流程，再将 MSA 交给结构模型推理。搜索阶段的资源特征与 GPU 推理不同：它涉及数据库扫描、CPU、多线程、主机内存和临时存储；结构推理才主要依赖 GPU 与显存。先明确是偶尔处理少量序列，还是需要批量、可控且可追溯地处理大量序列，再选择在线服务或本地数据库方案。拥有 GPU 不代表本地 MSA 数据库已经准备好。

在线服务适合快速探索，但输入会发送到外部服务；使用前确认序列和项目信息是否允许外传，并考虑服务可用性、排队、速率限制和结果留存。尚未公开或受合作协议限制的序列应采用单位批准的本地或受控服务。本地数据库要规划下载、索引、版本固定、磁盘容量、更新方式和并发读取；数据库变化可能改变 MSA 命中与最终结构结果。

批量输入要能追溯

FASTA 每条记录的 ID 应唯一、简单且稳定，序列不含空白和非法字符。不要在 ID 中放个人信息、保密项目名或路径。用 manifest 记录 query ID、序列来源、校验和、长度、运行日期、搜索模式和数据库版本；即使序列长度相同，也不能用文件顺序代替 ID 匹配结果。对重复序列去重时保存原始 ID 到新 ID 的映射，避免丢失样本对应关系。

固定批次大小，先用少量代表序列确认环境、数据库和输出命名，再扩大运行。每条 query 有明确状态：等待、MSA 完成、模型完成或失败。结构文件存在不代表搜索完整。失败重跑只针对失败 ID，并使用新日志或状态字段更新，避免覆盖已完成结果。

检查 MSA 内容而非只数行

MSA 深度、多样性和覆盖范围可能影响模型输入，但命中更多不一定更好。检查是否为空、同源序列是否过度相似、有效覆盖是否集中在局部、是否存在错配或污染。低同源、长插入或多结构域蛋白可能得不到理想 MSA；应报告限制，不能把低质量 MSA 的结果与高质量样本视为相同证据强度。

复用预先生成的 MSA 时，确认其对应同一条序列、同一数据库和同一处理参数，并保存 A3M 等原始文件、搜索日志和序列校验和。序列修改后不应默认沿用旧 MSA。按 query ID 组织独立目录，避免不同批次同名文件覆盖。

本地数据库的空间与并发规划

本地搜索数据库可能远大于 query FASTA，还需要索引和运行缓存。磁盘计划包含下载包、解压目录、数据库索引、临时目录、MSA 输出、模型结果和版本备份。多个进程同时扫描数据库时，增加 CPU 不一定按比例提速，还可能造成存储争用。先测小批次的数据库读取速度、主机 RAM、CPU、临时盘和单条 query 用时，再估计并发。

推理部分另记录 GPU 型号、显存峰值、序列长度、模型数、seed 数和耗时。不能把搜索阶段 CPU 负载当作 GPU 推理速度，也不能用短序列的显存需求推断长序列。批量较大时按序列长度分层估算，并先测试最长序列代表样本。

逐条验收模型输出

对每条序列确认 MSA 是否完成、模型是否正常退出、结构是否可解析、链和残基是否对应输入、日志是否报错。查看预测置信度和覆盖范围，对低质量或缺段结果单独标注。合成示例只能验证格式与运行入口，不能证明真实序列的 MSA 质量或结构可信度。

交付保留 query FASTA、manifest、MSA、软件版本、数据库版本、日志、结构和置信度摘要。共享前去除敏感路径和样本标识，并确认数据库与权重许可允许相应使用。服务器配置由真实批量规模、数据敏感性、并发和实测资源曲线共同决定。

区分缓存命中与新搜索

批量运行可能复用已有 MSA 或缓存，第二次执行比第一次更快并不代表搜索算法或硬件变快。记录每条序列是否命中缓存、缓存创建时间、数据库版本和输入序列校验和。缓存键若只使用短 ID 而 ID 在项目间重复，可能误取其他序列的结果；应确保 ID 唯一，必要时单独校验序列摘要。

清理缓存前确认哪些结果可重建以及是否已有备份。MSA 数据也可能包含外部数据库序列信息，按数据和许可要求管理。在线搜索结果与本地数据库搜索的命中集合可能不同，比较结构时说明模式差异，不能把两种搜索条件混成一个实验。

根据 MSA 结果选择是否复核

为每条 query 计算长度、MSA 序列数、有效覆盖比例或其他项目适用的深度指标，标记空 MSA、极浅 MSA 和覆盖断裂序列。阈值应根据研究对象和模型流程定义，不能直接套用一个通用序列数。对关键蛋白人工浏览 MSA，排查命中是否为同源家族成员、是否存在多结构域错配，以及对齐是否覆盖目标功能区域。

结构模型的预测结果和 MSA 是关联但不同的证据层。若模型置信度变化很大，可比较不同 MSA 来源或过滤规则，但需要固定结构模型版本、输入序列和其他设置。将差异归因于具体 MSA 条件，不要只记录最终结构分数。

把失败重跑控制在明确范围

如果只在搜索阶段失败，保留已完成的结构任务和对应 MSA 状态；如果 MSA 不完整，则不要让后续推理继续使用部分文件。按 query 状态生成待重跑列表，并将失败原因分成网络、数据库、格式、资源或模型阶段。每次重试记录新的日志路径，避免旧的空输出被新流程误识别为成功缓存。

报告批量完成率和失败序列清单，比只给成功结果数更有用。对关键序列应人工确认最终使用的 MSA 和结构版本，尤其是多次重试后数据库或参数曾变化的情况。

写清楚 MSA 来源与模型配置

论文或组会中至少说明使用在线还是本地搜索、数据库版本、搜索参数、MSA 过滤规则、ColabFold/模型版本、seed 数、输入序列版本和置信度检查方法。若某条序列没有可用 MSA，应明确标记并说明后续如何处理。不同来源的 MSA 不应在未记录的情况下混合，否则他人无法判断结构变化来自输入证据还是推理参数。

将这些信息存入每次运行的 manifest，而不是只写在个人笔记。若批量结果需要重新生成，固定输入校验和和数据库版本可避免同一 ID 被误配到更新后的序列。

明确共享服务的可用边界

在线或共享 MSA 服务可能有排队、速率限制、维护时间和数据库更新，课题组应保存实际运行日期与服务版本信息。对于需要长期复现的分析，记录搜索参数和原始 MSA 文件，避免只依赖远端服务日后再次返回相同结果。若服务条款限制数据类型或请求量，应先确认再提交批量任务。

服务暂时不可用时，将未处理 query 状态保留并按原设置恢复；不要悄悄改用另一个搜索来源后继续把结果作为同一批次。来源变化会影响 MSA 内容，应在结果表中明确区分。

确认结果和输入一一对应

批处理结束时自动比对查询清单与 MSA、模型输出清单，逐条列出缺失、重复和额外文件。文件名可以规范化，但映射表应保留原始 query ID。不要因为目录中结构文件总数接近输入数就忽略一条失败和一条重复输出；这种错配会把结构结果归到错误蛋白。

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

ColabFold官方批处理与MSA来源 · 来源资料

ColabFold批处理入口实现 · 来源资料

ColabFold本地MSA搜索入口 · 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。