用结构清单组织ProteinMPNN设计任务与可追溯输出

完整操作教程 · 参考固定 · 往返恢复验收

ProteinMPNN根据给定骨架设计序列。批量任务的关键是知道每份序列来自哪个结构、设计哪条链、使用什么模型和采样设置，而不是把所有PDB放到一个目录就算准备完成。本篇使用官方protein_mpnn_run.py的单PDB入口，先顺序处理结构清单。

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。

ProteinMPNN根据给定骨架设计序列。批量任务的关键是知道每份序列来自哪个结构、设计哪条链、使用什么模型和采样设置，而不是把所有PDB放到一个目录就算准备完成。本篇使用官方protein_mpnn_run.py的单PDB入口，先顺序处理结构清单。

冻结结构身份与设计链

环境已按官方说明安装并通过示例，模型权重与程序版本对应。输入PDB需有可解析的主链坐标、明确链ID和残基映射；多模型、插入码、缺失残基及非标准残基先审阅。设计链与固定链要按目标决定，不用文件名猜链，也不把缺失骨架自动当作可靠设计区域。

清单字段	本例含义	验收
job / pdb	唯一任务名与绝对PDB路径	保留原文件SHA
chains	单条待设计链，如A	在结构中确认存在
seed	大于零的固定随机种子	0在该工具中表示随机选种子
数量与batch	4条、batch_size=1教学值	数量须与实际FASTA核对

一个结构使用一个新目录

jobs.tsv无表头，四列为job、PDB绝对路径、单条链ID、正整数seed，TAB分隔。下面为Linux Bash；MPNN_REPO填固定commit的官方仓库，先激活兼容环境。示例每次只设计一条链，复杂固定位置、绑定位点或多链约束需按官方JSONL接口另定，不在清单中偷偷拼参数。

```
set -euo pipefail
: "${MPNN_REPO:?■■■■commit■ProteinMPNN■■■■}"
test -s "$MPNN_REPO/protein_mpnn_run.py"
mkdir mpnn-batch-001
root="$PWD/mpnn-batch-001"
git -C "$MPNN_REPO" rev-parse HEAD > "$root/commit.txt"
test "$(cat "$root/commit.txt")" = "8907e6671bfbfc92303b5f79c4b5e6ce47cdef57"
completed=0
while IFS=$'\t' read -r job pdb chain seed || [[ -n "$job$pdb$chain$seed" ]]; do
  [[ "$job" =~ ^[A-Za-z0-9_-]+$ ]]
  [[ "$chain" =~ ^[A-Za-z0-9]$ ]]
  [[ "$seed" =~ ^[1-9][0-9]*$ ]]
  [[ "$pdb" == /* ]]
  test -s "$pdb"
  out="$root/$job"
  mkdir "$out"
  sha256sum "$pdb" > "$out/input.sha256"
  /usr/bin/time -v -o "$out/run.time" python "$MPNN_REPO/protein_mpnn_run.py" \
    --pdb_path "$pdb" --pdb_path_chains "$chain" --out_folder "$out" \
    --num_seq_per_target 4 --batch_size 1 --sampling_temp 0.1 \
    --seed "$seed" --save_score 1 > "$out/run.log" 2>&1
  shopt -s nullglob
  fasta=("$out"/seqs/*.fa)
  test "${#fasta[@]}" -eq 1
  test "$(grep -c '^>' "${fasta[0]}")" -eq 5
  completed=$((completed+1))
done < jobs.tsv
test "$completed" -gt 0
```

五条记录不等于五条设计

本例一个温度、四次采样，FASTA通常包含一条原始序列加四条设计。首条标题记录设计链、固定链、模型、种子等信息，后续记录包含采样编号与分数。读取标题确认实际设计链与清单一致，再检查设计序列长度、残基范围和原始结构映射；仅靠grep计数不能验证链选择正确。

官方当前实现用num_seq_per_target整除batch_size组织批次，不可随意选择不整除组合后假定仍得到所填数量。多条设计链的序列会按约定以斜线分隔，不能将斜线当氨基酸或丢掉链边界。本例限制单链，正式扩展时应保留链到序列段的映射和对应长度。

资源与设计评价分开

程序按CUDA可用性选设备。需要CPU与GPU样例比较时，在独立结果目录保持同一输入、权重、采样设置和种子，分别控制可见设备并读取实际日志；不能仅从机器装有显卡判断本次用了GPU。采样batch增大可能提高显存需求，先测完整结构任务再决定。

保存模型权重校验值、Python/PyTorch版本、实际设备和命令；只记录Git commit还不足以复建全部环境。外层先顺序执行，再测少量结构并发，每项GPU任务须获得独立或明确共享的额度，不能让所有作业默认争用同一张卡。

ProteinMPNN分数来自模型序列概率，不是结合自由能或实验成功率。序列重复、结构相容性、功能约束、后续独立结构预测与实验评价另行安排。失败保留日志，在新目录重试；更换骨架、设计链或模型时建立新任务版本，不能覆盖旧FASTA后继续使用原任务标签。

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

ProteinMPNN官方接口与示例 · 来源资料

采样批次、设备与FASTA实现 · 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。