建立蛋白配体组合清单并批量验证预测输入与输出

完整操作教程 · 参考固定 · 往返恢复验收

Boltz批量蛋白—配体预测先建立组合身份，再构建模型输入。一个蛋白与十个配体是十个独立复合物任务，不能只用文件序号追踪来源。本篇使用当前Boltz-2结构预测接口，每项一条蛋白链和一个小分子，提供本地MSA；不默认运行亲和力模块，也不把模型置信度当成结合实验结论。

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。

Boltz批量蛋白—配体预测先建立组合身份，再构建模型输入。一个蛋白与十个配体是十个独立复合物任务，不能只用文件序号追踪来源。本篇使用当前Boltz-2结构预测接口，每项一条蛋白链和一个小分子，提供本地MSA；不默认运行亲和力模块，也不把模型置信度当成结合实验结论。

把蛋白、配体和MSA放进一张清单

pairs.csv带表头job、protein_fasta、smiles、msa，路径填写绝对路径。先确认蛋白序列、构建体边界和质子化/互变异构及立体化学选择，配体ID不能只写商品名。SMILES列中的逗号必须按CSV规则引用。每项的MSA应与其蛋白序列匹配，不因长度相同就共用另一个蛋白的MSA。

字段	本例规则	边界
protein_fasta	一条标准氨基酸序列	变体或截短即新任务
smiles	可解析且立体化学经审阅	解析成功不等于化学状态正确
msa	已验收的本地a3m绝对路径	不把msa: empty当资源优化
模型与输出	显式boltz2、一个结构样本	增加采样会改变开销和结果集合

从组合表生成独立YAML

需要Python 3、Biopython、PyYAML与RDKit。以下只创建输入，不提交网络MSA请求。RDKit规范化字符串并保留原清单，不能替代对配体化学状态的判断。输出目录必须全新，重复任务名、非法序列或不可解析SMILES会停止准备。

```
import csv,re,json
from pathlib import Path
import yaml
from Bio import SeqIO
from rdkit import Chem
jobs=[];seen=set()
with open("pairs.csv",newline="") as f:
    reader=csv.DictReader(f)
    assert {"job","protein_fasta","smiles","msa"}<=set(reader.fieldnames or [])
    for row in reader:
        job=row["job"];assert re.fullmatch(r"[A-Za-z0-9_-]+",job) and job not in seen
        seen.add(job)
        fasta=Path(row["protein_fasta"]);msa=Path(row["msa"])
        assert fasta.is_absolute() and msa.is_absolute() and msa.is_file() and
msa.stat().st_size>0
        records=list(SeqIO.parse(fasta,"fasta"));assert len(records)==1
        seq=str(records[0].seq).upper();assert seq and set(seq)<=
set("ACDEFGHIKLMNPQRSTVWY")
        mol=Chem.MolFromSmiles(row["smiles"]);assert mol is not None and
mol.GetNumAtoms()>0
        ligand=Chem.MolToSmiles(mol,isomericSmiles=True)
        obj=
{"version":1,"sequences":[{"protein":{"id":"A","sequence":seq,"msa":str(msa)}},
             {"ligand":{"id":"B","smiles":ligand}}]}
        jobs.append((job,obj,row))
assert jobs
root=Path("boltz-input-001");root.mkdir();(root/"inputs").mkdir()
for job,obj,row in jobs:
    (root/"inputs"/(job+".yaml")).write_text(yaml.safe_dump(obj,sort_keys=False))
```

```
(root/"source-plan.json").write_text(json.dumps([r for j,o,r in jobs],indent=2))
print("complexes",len(jobs))
```

用固定模型处理输入目录

先在已验证的Boltz环境执行boltz predict --help并保存安装版本。下例一个GPU设备、两条预处理线程和一个结构样本均是教学值，不是通用规格。若需要比较CPU，用独立新目录改变accelerator并核对兼容条件；不能从支持CPU推导任何任务都适合CPU长时间运行。

```
set -euo pipefail
mkdir boltz-run-001
python -c 'from importlib.metadata import version; print(version("boltz"))' > boltz-run-
001/version.txt
boltz predict --help > boltz-run-001/help.txt
/usr/bin/time -v -o boltz-run-001/run.time boltz predict boltz-input-001/inputs \
  --model boltz2 --accelerator gpu --devices 1 --preprocessing-threads 2 \
  --diffusion_samples 1 --seed 101 --out_dir "$PWD/boltz-run-001" \
  > boltz-run-001/run.log 2>&1
for yaml in boltz-input-001/inputs/*.yaml; do
  job=$(basename "$yaml" .yaml)
  pred="boltz-run-001/boltz_results_inputs/predictions/$job"
  test -s "$pred/${job}_model_0.cif"
  test -s "$pred/confidence_${job}_model_0.json"
done
```

按组合验收，别只看整批退出

逐项读取CIF和confidence JSON，核对蛋白、配体、链映射、坐标与模型数；为清单中的每个job登记通过、失败或未运行。预测目录名由输入目录stem派生，本例是boltz_results_inputs，不能只照文档示意图寻找OUT/predictions。文件存在后仍需确认可解析及数值有限。

检查配体是否完整、化学键与立体状态是否合理，以及局部接触、碰撞和蛋白构象。置信度用于模型自身评估，不是实验亲和力；需要亲和力预测时另核其支持对象与限制，并单独保存任务条件和字段解释。本例没有properties.affinity，因此不应要求出现affinity JSON。

保存YAML、原始组合表、蛋白/MSA/权重指纹、程序版本和缓存位置。修改输入后建立新输出目录，避免默认缓存复用让旧预测混入新组合。MSA、预处理、推理和结果检查分别测量，记录主机RAM及GPU显存；先完成代表组合，再安排符合显存与工作盘额度的小批次，不承诺固定结构长度对应固定显存。

相同蛋白在不同配体任务中可引用同一份冻结MSA，但必须保留query序列与生成协议。正式提交前逐项核对MSA首序列、蛋白构建体和配体身份，选择几个YAML人工检查，确认没有因CSV列错位让同一个配体被反复提交。大规模组合还应记录实际计划数与已生成YAML数。

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

Boltz官方模型与输入说明 · 来源资料

YAML与预测参数 · 来源资料

实际输出目录及CLI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。