建立突变清单并调度Rosetta稳定性计算与日志核对

完整操作教程 · 参考固定 · 往返恢复验收

大量突变稳定性计算先明确协议与编号，再调度CPU。把每条突变写成一个任务，保留野生型结构、突变文件和随机状态，才能判断失败或异常分数来自哪里。本篇使用当前cartesian_ddg的非legacy单体路线，前提是已按匹配协议完成笛卡尔预松弛，不从原始PDB直接开始批量比较。

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。

大量突变稳定性计算先明确协议与编号，再调度CPU。把每条突变写成一个任务，保留野生型结构、突变文件和随机状态，才能判断失败或异常分数来自哪里。本篇使用当前cartesian_ddg的非legacy单体路线，前提是已按匹配协议完成笛卡尔预松弛，不从原始PDB直接开始批量比较。

突变位置必须对应Rosetta pose

先核软件使用条件、版本及数据库，环境通过官方单突变示例。建立实验编号、PDB链/残基号和Rosetta pose连续编号的对应表，并核对原氨基酸。插入码、缺失残基和清理步骤会改变映射，不能将文献中的位置数字原样填入mut_file。这里只讨论单体稳定性，不将其冒充界面结合能。

材料	本例约定	验收
relaxed.pdb	与计算相同分数函数及fa_max_dis完成预松弛	保存原始与处理后映射
mutations.tsv	任务名、WT字母、pose位置、突变字母、seed	一项一个单点突变
mutations.mut	每项独立文件	多个突变写进一组可能变为联合突变
结果	WT与MUT多轮能量	不只看一个最小值

从清单生成单突变文件

mutations.tsv无表头、TAB分隔，例如job001、A、42、V、101；此例编号只演示格式，并不对应任何真实蛋白。先人工或用本版Rosetta映射核对清单中WT与pose位置，再运行下列Python。它只做语法与目录检查，不声称已核对结构中的原氨基酸。

```
import csv,re
from pathlib import Path
rows=list(csv.reader(open("mutations.tsv"),delimiter="\t"));assert rows
seen=set();parsed=[];aa=set("ACDEFGHIKLMNPQRSTVWY")
for row in rows:
    assert len(row)==5
    job,wt,pos,mut,seed=row
    assert re.fullmatch(r"[A-Za-z0-9_-]+",job) and job not in seen
    assert wt in aa and mut in aa and wt!=mut
    assert pos.isascii() and pos.isdigit() and int(pos)>0
    assert seed.isascii() and seed.isdigit() and int(seed)>0
    seen.add(job);parsed.append((job,wt,int(pos),mut,int(seed)))
root=Path("ddg-input-001");root.mkdir()
for job,wt,pos,mut,seed in parsed:
    d=root/job;d.mkdir()
    (d/"mutations.mut").write_text(f"total 1\n1\n{wt} {pos} {mut}\n")
    (d/"seed.txt").write_text(str(seed)+"\n")
```

每条突变在自己的工作目录执行

DDG、ROSETTA_DB、RELAXED均填匹配版本的绝对路径。以下按官方当前非legacy入口设置，五轮为教学试跑值；实际采样、收敛与预松弛协议应预先确定。当前源码以mut_file基本名生成结果，本例为mutations.ddg和mutations.json；旧文档中的PDB前缀示意不能用于盲目验收。

```
set -euo pipefail
: "${DDG:?cartesian_ddg■■■■■■■■■■}"
: "${ROSETTA_DB:?■■■■■■■■■■}"
: "${RELAXED:?■■■■■■■■■■PDB■■■■■}"
test -x "$DDG"
test -s "$RELAXED"
test -d "$ROSETTA_DB"
mkdir ddg-run-001
root="$PWD/ddg-run-001"
for input in ddg-input-001/*; do
  test -d "$input"
  job=$(basename "$input")
  mkdir "$root/$job"
  cp "$input/mutations.mut" "$root/$job/"
  seed=$(cat "$input/seed.txt")
  [[ "$seed" =~ ^[1-9][0-9]*$ ]]
  (
    cd "$root/$job"
    sha256sum "$RELAXED" mutations.mut > inputs.sha256
    /usr/bin/time -v -o run.time "$DDG" -database "$ROSETTA_DB" -s "$RELAXED" \
      -ddg:mut_file mutations.mut -ddg:iterations 5 -ddg:legacy false -ddg:json true \
      -ddg:cartesian -ddg:dump_pdbs false -ddg:bbnbrs 1 -ddg:frag_nbrs 2 \
      -ddg:flex_bb false -ddg:force_iterations false -ddg:score_cutoff 1.0 \
      -score:weights ref2015_cart -fa_max_dis 9.0 \
      -run:constant_seed -run:jran "$seed" > run.log 2>&1
    test -s mutations.ddg
    test -s mutations.json
    python -m json.tool mutations.json > json-check.txt
  )
done
```

复核异常，再汇总稳定性差值

逐任务核对实际突变、WT与MUT轮次、数值有限性及日志收敛；允许协议提前收敛，不机械要求一定五行。按选定协议计算MUT与WT的能量差及采样波动，保留原始能量。Rosetta能量单位与实验自由能并非同一尺度，不未经校准直接报告成可靠kcal/mol或实验稳定性结论。

突变后异常高分先检查编号、WT、局部缺失和预松弛条件；缺JSON先核编译是否具备当前非legacy路线要求。更改分数函数、柔性范围或原结构后需新任务版本。独立重复使用不同种子，不能把同一种子重跑当成新增采样。

先顺序测代表突变完整任务，记录墙钟、RSS和临时盘，再增加外层并发。本例没有通用GPU或线程加速参数，多个任务各自工作目录可避免结果续写与缓存混用。整批表保留失败、未运行和待复核项，不只导出最稳定的候选；实验优先级还需结合方法适用性与独立证据。

同一位点的多个替换分别占一行，双突变或组合突变另定义任务，不能把清单打包为一个mut_file后仍按独立单点解释。保存实验编号到pose编号的映射与核对日期，结构重新清理或预松弛流程改变后再次核验，不复用来源不明的数字位置。

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

Rosetta当前Cartesian ddG协议与限制 · 来源资料

当前非legacy实现与实际输出名称 · 来源资料

当前参数定义 · 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。