为多轨迹分析建立文件配对、分块计算与结果检查

完整操作教程 · 参考固定 · 往返恢复验收

多条轨迹计算RMSD和接触，首先要知道每条坐标对应哪个拓扑及原子顺序。原子数相同仍可能配错体系。本篇用MDAnalysis 2.11接口逐帧读取，按千帧拆分CSV结果；计算蛋白Cα相对本轨迹首帧的拟合RMSD，以及指定两组Cα之间的距离接触数。

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。

多条轨迹计算RMSD和接触，首先要知道每条坐标对应哪个拓扑及原子顺序。原子数相同仍可能配错体系。本篇用MDAnalysis 2.11接口逐帧读取，按千帧拆分CSV结果；计算蛋白Cα相对本轨迹首帧的拟合RMSD，以及指定两组Cα之间的距离接触数。

先确定配对与周期边界处理

trajectories.tsv无表头，三列为唯一任务名、拓扑路径、轨迹路径。它们应来自同次体系，原子身份与顺序已验收；特别是去水、只保留蛋白或重新排序后的轨迹，必须有匹配拓扑。RMSD前蛋白应已按适合本体系的方法处理为完整分子，不能让跨周期边界的断裂坐标直接进入拟合。

指标	本篇定义	解释边界
RMSD	蛋白Cα拟合到本轨迹首帧	不同首帧并非同一公共参考
接触	两组Cα距离不超过8 Å的原子对数	不是氢键、结合能或native Q
周期边界	接触按有效盒参数取最小镜像	不能替代RMSD所需完整分子
分块	每个CSV最多1000帧	逐帧坐标，不加载整条轨迹

先选择两个不重叠的接触组

环境需要MDAnalysis、NumPy及可读取本项目格式的依赖。GROUP_A、GROUP_B填写实际选择表达式，例如protein and name CA and resid 1:10与另一个存在的片段；编号不是通用生物学区域。下面存为analyze_trajectories.py后运行，使用默认Å与ps单位，不能与GROMACS常见nm输出混用。

```
import csv,os,re,json,math
from pathlib import Path
import numpy as np
import MDAnalysis as mda
from MDAnalysis.analysis.rms import rmsd
from MDAnalysis.lib.distances import capped_distance
sel_a=os.environ["GROUP_A"];sel_b=os.environ["GROUP_B"]
jobs=list(csv.reader(open("trajectories.tsv"),delimiter="\t"));assert jobs
assert all(len(row)==3 for row in jobs)
assert len({row[0] for row in jobs})==len(jobs)
root=Path("trajectory-analysis-001");root.mkdir()
for job,top,traj in jobs:
    assert re.fullmatch(r"[A-Za-z0-9_-]+",job)
    u=mda.Universe(top,traj,in_memory=False)
    ca=u.select_atoms("protein and name CA")
    a=u.select_atoms(sel_a);b=u.select_atoms(sel_b)
    assert len(ca)>=3 and len(a)>0 and len(b)>0
    assert not set(a.indices)&set(b.indices)
    assert len(a)*len(b)<=1_000_000,"■■■■■■■■■■■■■■■■■■"
    u.trajectory[0];reference=ca.positions.copy()
    assert np.isfinite(reference).all()
    d=root/job;d.mkdir();handle=None;last=-math.inf;count=0
    try:
        for i,ts in enumerate(u.trajectory):
            assert math.isfinite(float(ts.time)) and ts.time>last
            last=float(ts.time)
            assert np.isfinite(ca.positions).all()
            assert np.isfinite(a.positions).all() and np.isfinite(b.positions).all()
```

```
            box=ts.dimensions
            assert box is not None and np.isfinite(box).all()
            assert np.all(box[:3]>0) and np.all((box[3:]>0)&(box[3:]<180))
            value=float(rmsd(ca.positions,reference,center=True,superposition=True))
            assert math.isfinite(value)
            pairs=capped_distance(a.positions,b.positions,max_cutoff=8.0,
                                  box=box,return_distances=False)
            if i%1000==0:
                if handle is not None:handle.close()
                handle=(d/f"part_{i//1000+1:05d}.csv").open("x",newline="")
                writer=csv.writer(handle);writer.writerow(["frame","time_ps","CA_RMSD_
A","CA_contacts_le8A"])
            writer.writerow([ts.frame,ts.time,value,len(pairs)]);count+=1
    finally:
        if handle is not None:handle.close()
    assert count>0
    (d/"status.json").write_text(json.dumps({"status":"basic_metrics_
written","frames":count,
        "topology":str(Path(top).resolve()),"trajectory":str(Path(traj).resolve()),
        "group_a":sel_a,"group_b":sel_b,"MDAnalysis":mda.__version__},indent=2))
```

核对分块结果与指标含义

每条轨迹应有status和连续编号的CSV，汇总数据行数应等于status.frames，首末时间与输入读取结果对应。2001帧应分成1000、1000和1行数据，表头不计帧数。抽查首帧RMSD接近零、一个接触原子对及一个无接触帧，确认距离阈值和单位正确。

同一轨迹的千帧块只是存储分块，不能当成独立重复做统计。多轨迹若要比较公共参考，另提供匹配结构和一致的原子选择；本例每条使用自己的首帧，不能直接把差异解释为相对同一个实验结构的偏离。接触计数还会受所选原子数和区域影响，比较时保持定义一致。

排错和批量扩展

遇到拓扑/坐标原子数不符、空选择、重复时间或无效盒参数时停止，不删除断言继续计算。输出块可能已经存在但status尚未生成，应保留为失败现场；在新目录重跑，不混入旧块。检查器无法仅靠XTC坐标证明化学身份正确，来源配对与原子映射仍是必要前提。

先顺序处理一条完整代表轨迹，测读盘、CPU和峰值内存，再增加外层并发。每项拥有自己的Universe与输出目录，避免线程共享一个可变的轨迹游标。大量轨迹同时扫描可能受存储带宽限制；分块写结果降低结果缓存，不代表计算与拓扑索引完全不占内存。保留选择、版本、输入指纹及处理规则，汇总时登记失败和未运行项。

输入时间重复时先核接续段是否包含重叠帧，不能直接去重后隐藏原始处理过程。接触阈值或选择表达式变更时建立新的分析版本，保留各块的帧数和选择原子数量，使不同结果具有可比较的定义。

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

MDAnalysis RMSD及单位、周期边界提示 · 来源资料

MDAnalysis距离计算与周期盒 · 来源资料

MDAnalysis轨迹读取与选择 · 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。