按窗口组织自由能计算的输入、执行清单和完整性检查

完整操作教程 · 参考固定 · 往返恢复验收

自由能计算的多个lambda窗口可以独立运行，但全部窗口完成也不等于自由能已经收敛。本篇以GROMACS2026.4、三个已论证的离散状态和两个独立重复为教学任务表，组织六份已生成并验收的TPR，核对缺失结果和

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。

自由能计算的多个lambda窗口可以独立运行，但全部窗口完成也不等于自由能已经收敛。本篇以GROMACS2026.4、三个已论证的离散状态和两个独立重复为教学任务表，组织六份已生成并验收的TPR，核对缺失结果和时间序列，不提供通用于所有体系的lambda间距。

先把状态、重复与TPR对应

准备tasks.tsv，表头为state、replica、tpr，TAB分隔。示例状态0、1、2对应本项目已批准lambda向量的零基索引，不是lambda数值本身；每个状态都有R1/R2且使用各自TPR。TPR的拓扑A/B、软核、温压、采样时长、dhdl设置和初始化已完成单任务验证。

层级	本例约定	验收
状态	0、1、2三个索引	gmx dump核对实际lambda设置
重复	R1、R2	初始化/随机种子来源明确
任务	每个状态×重复一份TPR	不能复用同份TPR冒充另一状态
方法	独立固定窗口	不代替副本交换或expanded ensemble
资源	先CPU或已核验GPU路线	按整套方法与日志确认设备

按显式清单顺序执行

代码保存为run_windows.py。NB_MODE填cpu或gpu；只有已验证当前自由能模式支持候选GPU时才设置gpu及GPU_SUPPORT_VERIFIED=1，这只是运行前项目确认，不是程序替你检测兼容。两线程为演示。设备可见范围由调度环境决定，不自动占用其他GPU。

```
import csv,os,subprocess,math
from pathlib import Path
with open("tasks.tsv",newline="") as f:rows=list(csv.DictReader(f,delimiter="\t"))
expected={(str(i),r) for i in range(3) for r in ["R1","R2"]}
pairs=[(r["state"],r["replica"]) for r in rows]
assert len(pairs)==len(expected) and set(pairs)==expected
paths=[Path(r["tpr"]) for r in rows]
assert all(p.is_absolute() and p.is_file() and p.stat().st_size>0 for p in paths)
assert len({p.resolve() for p in paths})==len(paths)
mode=os.environ.get("NB_MODE","cpu");assert mode in {"cpu","gpu"}
if mode=="gpu":assert os.environ.get("GPU_SUPPORT_VERIFIED")=="1"
root=Path("lambda-batch-001");root.mkdir()
for row,path in zip(rows,paths):
    d=root/f"s{row['state']}_{row['replica']}";d.mkdir()
    with (d/"tpr.dump.txt").open("w") as out:
        subprocess.run(["gmx","dump","-s",str(path)],stdout=out,check=True)
    with (d/"run.stdout.txt").open("w") as out:
        subprocess.run(["/usr/bin/time","-v","-o","resources.txt","gmx","mdrun",
            "-s",str(path),"-deffnm","run","-dhdl","dhdl.xvg", "-nb",mode,
            "-ntmpi","1","-ntomp","2"],cwd=d,stdout=out,stderr=subprocess.STDOUT,check=
True)
    previous=None;count=0
    with (d/"dhdl.xvg").open() as f:
        for line in f:
            if not line.strip() or line.startswith(("#","@")):continue
            values=list(map(float,line.split()));assert len(values)>=2
            assert all(math.isfinite(v) for v in values)
```

```
            assert previous is None or values[0]>previous
            previous=values[0];count+=1
    assert count>=2
    assert (d/"run.log").is_file()
    (d/"FILES_CHECKED").write_text("verify expected endpoint and sampling before
scientific acceptance\n")
```

检查每个任务的实际终点

逐项读取tpr.dump与run.log，核对init-lambda-state、方法、实际步数、终止原因及设备分工；dhdl的最后时间应符合本协议输出频率和目标时段。FILES_CHECKED只表明基本文件和有限值检查通过，不表示达到生产长度，更不是收敛标记。缺一必要窗口就在任务表标未完成。

不同窗口的平衡段、相关时间和有效样本量可能不同；后续BAR或MBAR应核对对应能量差列、状态重叠及独立重复一致性。不能将所有dhdl按文件名拼成一条时间序列，也不直接平均各窗口自由能数字。对于重复轨迹还要区分独立重复与同一轨迹的分段。

本例任务数等于三状态乘两重复，实际项目按完整lambda向量生成预期集合。更改向量或新增中间状态后重新建立TPR及清单，保存版本和参数差异；不要把一个旧窗口文件复制到新状态目录填齐数量。只有状态、重复、数据和分析方法都对应，才进入估计阶段。

资源增加解决吞吐，不能替代采样

先测代表窗口的完整耗时、RAM、显存、检查点和轨迹共存空间，再试获配范围内的小并发。不同lambda状态可能有不同GPU支持或性能约束，CPU与GPU路线应分别验收；不能因很多窗口可并行，就认定全流程都由GPU加速。

失败后保留本任务日志与检查点，依据相同TPR和兼容版本恢复；需要修改协议时另建尝试目录，不覆盖旧dhdl。汇总全部任务的成功范围、缺失项与采样诊断，计算完成、文件完整和科学收敛三项分别记录，资源评估据真实端到端结果进行。

任务表中的TPR还应保留字节校验值，防止重命名或复制后状态标识与实际内容分离。

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

GROMACS2026.4自由能状态与lambda参数 · 来源资料

GROMACS2026.4 mdrun/dhdl/GPU选项 · 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。