Dandelion队列分析教程：固定依赖、分批构图与资源记录

完整操作教程 · 参考固定 · 往返恢复验收

明确Dandelion后端与已整理BCR输入，把距离计算、图和布局分开记录，避免随机拆网络后误称等价。

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。

Dandelion大型BCR网络的资源，既受细胞数影响，也受克隆规模、序列距离和布局方法影响。先确定要构建什么网络，再讨论分批与硬件。把同一克隆随机拆到多个文件，分别建图后简单拼接，并不保证得到原本的全体网络。

先锁定后端和输入阶段

当前官方API区分base与其他后端，本篇明确使用base接口及已整理的.h5ddl对象。输入已完成合适的重注释、质量处理和克隆定义，包含有效细胞ID、clone_id及sequence_alignment_aa。它不是从原始VDJ文件自动补齐全部上游的脚本。

环境按官方安装说明准备并记录软件、数据库和依赖版本。重注释可能需要IgBLAST、Change-O等外部工具与参考，Python能import不证明这些步骤齐全；纯图阶段也不必为所有任务强行加GPU。容器需固定发布和挂载，不只记latest。

阶段	记录	分批边界
预处理与克隆	物种、参考、序列及规则	科学定义先确定
距离	键、clone/full模式、实际规模	不能任意切断同一分析集合
图构建	节点、边及单例处理	不同图范围分别验收
布局	算法、随机性、是否需要	可先不布局，不能当距离结果更准确

先建立不做布局的构图基线

下面以独立分析集合prepared.h5ddl为例，使用已核对的当前base API，显式按克隆内距离构图、单CPU并关闭布局。旧版本若没有这些接口，应按其文档重建兼容代码，不把另一后端的对象强行传入。

```
from pathlib import Path
import inspect, json
from importlib.metadata import version
from dandelion.base.io import read_h5ddl
from dandelion.base.tools import generate_network

required = {"clone_key", "distance_mode", "compute_layout", "n_cpus", "use_existing_
graph"}
assert required <= set(inspect.signature(generate_network).parameters)
vdj = read_h5ddl("prepared.h5ddl")
assert {"cell_id", "clone_id", "sequence_alignment_aa"} <= set(vdj.data.columns)
assert vdj.data.index.is_unique and vdj.metadata.index.is_unique
assert vdj.data["cell_id"].notna().all()
assert vdj.data["clone_id"].notna().all()
out = Path("dandelion-pilot")
out.mkdir()
generate_network(vdj, clone_key="clone_id", distance_mode="clone",
                 compute_layout=False, use_existing_graph=False, n_cpus=1, random_state=
101)
assert vdj.graph is not None
stats = [{"nodes": g.number_of_nodes(), "edges": g.number_of_edges()}
         for g in vdj.graph if g is not None]
assert stats
(out / "graph-stats.json").write_text(json.dumps(stats, indent=2), encoding="utf-8")
vdj.metadata.to_csv(out / "cell-metadata.csv")
vdj.write(str(out / "network.h5ddl"))
(out / "package-version.txt").write_text(version("sc-dandelion") + "\n", encoding="utf-
8")
```

先核对克隆、节点、边与输入细胞对应，再单独测试需要的布局。图中的距离和边由所选序列与方法决定，不是实验谱系时间或抗原特异性证明；布局相近也不说明两个处理协议等价。默认完整图和裁剪图的范围应分别说明。

分批按分析设计，而非只按文件大小

不同供者或已定义独立集合可以分别运行并保留清单，但跨时间或组织需要共同追踪的克隆，应保留所需关联。若目的是整个队列的网络，先评估工具支持的距离存储和分块计算，不把独立子图冒充完整图。sample参数会重采样，不能暗中用它替代全量资源测量。

当前部分接口支持lazy与Zarr相关设置，但它们会引入工作盘、侧文件和任务配置；不能只保存主对象就认为依赖完整。示例走非lazy基线，不据此保证大对象低内存。memory_limit等参数也应按其具体作用域理解，不当整个项目RAM硬上限。

保存阶段证据与失败范围

分别测预处理、距离、构图、布局和写出的墙钟、作业RAM及盘高水位。只有旧图被复用、跳过布局或样本被抽小的运行，不能与全量冷启动直接比速度。增加n_cpus前核对额度和总内存，先以单任务有效结果建立基线。

缺序列或clone列先修输入阶段，后端或参数不识别先匹配版本，磁盘与RAM失败分别查证。失败留目录和日志，新尝试保持方法与输入一致；交付对象、元数据及外部依赖清单并复读核对后，才算这组构图结果可继续使用。

本例显式关闭use_existing_graph，强制重新构图；输入对象即使带旧graph，也不把复用旧图的耗时当成本轮构图基线。

先确认克隆和细胞注释可用

Dandelion流程依赖一致的受体链、细胞条形码和元数据。运行前核对contig表的链类型、生产性标记、CDR3字段、样本前缀和细胞ID，并确认输入确实是同一套过滤策略。缺少某列时不要猜默认值补齐；回到原始注释文件或转换步骤确认语义。RNA和VDJ数据来自不同目录时，用稳定键建立对应表并抽样核验匹配。

克隆分配结果受克隆定义、距离规则和样本边界影响。多样本共同运行时，明确是否允许跨样本克隆比较以及这些样本是否处于同一分析设计。一个clone标签是算法定义的关系，不等价于确认的共同祖先或抗原特异性。对重要结论应并列展示链、序列、样本和克隆规则，并考虑合理参数变化对分组的影响。

图结构、缓存与内存

Dandelion可为细胞关系或克隆关系构建图。重用已有图能节约时间，但必须确认图对应完全相同的细胞集合、边定义和版本；若输入或参数改变，就应重建或显式验证图散列，不能把缓存图自动视为当前结果。本文例子以新图为基线，便于明确记录本次输入和输出关系。

图稀疏不意味着后续布局或转换步骤一定低内存。布局算法可能创建浮点坐标和临时副本，大图可先在子集或代表性样本上评估计算量，但子集布局不等于全数据的最终图。记录节点数、边数、非零元素、图生成时间、布局时间和峰值RSS，区分图构建、聚类、布局以及绘图耗时。若错误只发生在布局阶段，优先检查布局方法和规模，不要误判为前置克隆算法失败。

分层处理和Zarr的适用边界

将数据按样本或克隆分层处理可以控制任务范围，但拆分边界会影响跨样本连接。若生物学问题需要不同样本之间共享克隆信息，应先定义跨样本识别规则，再决定能否分批。分层后合并图时保留原始节点ID、样本前缀和边来源，确保不会把不同文库中相同裸barcode连成同一细胞。

基于Zarr或lazy对象的方案可能要求特定输入格式、存储后端和软件版本，并不意味着所有中间矩阵都会惰性计算。先按官方文档验证哪些步骤支持lazy、哪些步骤会materialize数据，并在小样本监测实际内存。若某阶段将对象整体加载进内存，仍需按峰值资源评估。不能只把AnnData写成Zarr就宣称可以无上限扩展。

结果验收与图形解释

结果检查包含图节点数与输入细胞数对应、边权和方向有意义、克隆分组没有非预期跨样本连接，以及图中代表性细胞与原始标签一致。保存图形参数、布局种子和节点ID映射。视觉布局受算法、随机种子和坐标变换影响；图上距离并不天然代表遗传距离，除非对应方法明确定义。

若图为空或节点大量缺失，先检查输入链过滤、barcode匹配和图构建参数；若图过密或运行缓慢，先检查边阈值和数据规模，但任何阈值变化都应作为方法调整并比较结果。报告中描述可支持的关系类型，不把网络连边直接写成细胞谱系因果关系。保留未连接节点，这些节点可能反映数据不足，而不是应被静默丢弃。

新旧图比较应看结构证据

重建图后，可比较节点集合、边数、连通分量、度分布和代表性节点的连接，而不只比较布局坐标。布局受种子和迭代过程影响，坐标移动不一定表示图关系变化；相反，看起来相似的布局也可能包含不同边。对关键图保存节点ID顺序和边表散列，作为是否可复用旧图的可验证条件。

图中节点或边异常时，从输入cell ID映射、克隆定义、过滤参数和图构建阈值逐层检查。记录每一步保留的细胞数和边数，能定位是输入过滤还是图计算造成变化。低度或孤立节点可保留并单独解释，不应为了画面紧凑而一概删除，否则会改变总体图结构和对边界细胞的观察。

如何解释不同分析粒度

按克隆、样本或细胞类型分层可让图更易读，但每一种粒度定义不同的问题。克隆关系图的边不代表细胞转录相似，表达邻接图也不等价于受体序列相似。作图标题、图例和报告应说明节点、边、距离及过滤规则，并避免把不同图层的连接关系混为同一种生物关系。

若比较两个条件的网络，先确保使用相同节点定义、克隆规则和边阈值。可分别报告节点覆盖、共享边、连通分量和参数敏感性；如节点集合不同，说明比较是基于各自网络还是共同交集。不得因为某种阈值产生更明显分组就只展示该阈值而隐去其他合理设置。

服务器资源与团队协作

共享环境中如需运行大图，先查看项目获得的内存、CPU和临时存储额度，确认Zarr目录读写权限和备份策略。多用户同时读写同一Zarr或缓存目录会有一致性风险，应为每个运行分配独立工作目录，最终经过核验再发布只读结果。不要把所有用户目录设成可写公共缓存来追求方便。

资源申请说明中列出节点数、边规模、是否计算布局、预期并发和存储保留期限。若瓶颈是布局，增加RAM不一定有帮助；若瓶颈是Zarr I/O，网络文件系统延迟也可能影响时间。分别记录CPU、内存和I/O等待，依据实测选择运行方案，并在报告中说明实际环境而非仅写“服务器完成”。

一次小样本试跑的复核表

选择一个具有代表性的样本，记录输入细胞数、contig数、克隆数、图节点/边数量、计算时间和峰值内存。检查每个输入细胞在结果中是否有明确状态，确认克隆分组没有超出预设样本范围。小样本只用于评估接口与资源，不应把网络形态直接推广为完整队列的生物学结论。

跑通后将参数和环境锁定，再扩大数据。若全量任务在某阶段失败，先与小样本记录对比该阶段输入规模和资源占用；如果规模增长导致峰值明显上升，测试按科学设计分层或调整图方法。每项调整后都重新检查节点覆盖和边定义，不能把仅仅“能跑完”作为唯一验收标准。

结果交付的最小内容

交付包包括克隆/节点表、边表或图对象、输入细胞映射、克隆规则、布局参数、软件版本、运行日志和结果说明。对节点ID和样本ID使用稳定命名，并保留从最终图回溯原始contig或细胞的路径。可视化图片应附图例，说明坐标由何种布局产生及哪些边被省略。

若结果图仅用于探索，标注探索性用途并保存完整数据表供审阅。用图形阈值筛掉边或细胞时，要记录筛选前后数量。共享网络图前确认序列或样本信息不会暴露受限数据；必要时仅共享匿名化的汇总关系并遵循项目审批。

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

Dandelion官方环境与前处理范围 · 来源资料

Dandelion当前后端API · 来源资料

base generate_network参数及模式 · 来源资料

base read_h5ddl输入 · 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。