Squidpy多切片邻域图：坐标分组、稀疏度与内存实测

用kNN接口按切片建图，核对坐标和邻居定义，并直接测量CSR矩阵的边数与字节数；不从h5ad大小或稠密矩阵公式猜图内存。

空间邻域图的边由坐标和建图规则决定。多切片任务若坐标重叠却未按library隔开，会出现跨片假邻居；若坐标单位或变换不一致，边长也无法解释。本文只从坐标表构造图，不载入表达矩阵，适用于需要先评估图结构和内存的场景。稀疏小切片也要逐片检查。

先核坐标和图定义

项目	检查方法	常见影响
观测ID	唯一；跨片重复时加片名	错配或覆盖行
x、y	有限数值，记录单位与变换	错单位会改变邻接关系
library_id	每点对应真实切片	避免切片互连
k或半径	符合平台几何与研究问题	kNN会跨越空洞找远邻

kNN固定每个点的近邻数，不设最大距离；点云有空洞、边缘或断裂时可能连出很长的边。需要物理距离上限时选择radius；规则Visium网格可采用grid定义。坐标重复先查配准和精度舍入，不能为消除警告就自动删点。Squidpy 1.8把旧spatial_neighbors标为弃用，本文使用spatial_neighbors_knn；运行前仍须核本机版本文档。

按切片建图并测CSR占用

coordinates.csv需要cell、x、y、library_id四列。library_id设为分类变量并传入library_key，Squidpy才会分片构图。先用n_jobs=1建立基准；并行只对多library建图生效，增加并发会增加瞬时内存且速度不一定线性提高。

```
import json, time
from pathlib import Path
import anndata as ad
import numpy as np, pandas as pd, squidpy as sq
from scipy.sparse import csgraph

m = pd.read_csv("coordinates.csv", dtype={"cell":str,"library_id":str})
cols = ["cell","x","y","library_id"]
assert set(cols) <= set(m) and not m[cols].isna().any().any()
assert m.cell.is_unique and m.library_id.str.len().gt(0).all()
xy = m[["x","y"]].to_numpy(float)
assert np.isfinite(xy).all()
k = 6  # ■■■■■■■■■■■■■■■■■
assert (m.groupby("library_id").size() > k).all()
obs = m.set_index("cell")[["library_id"]].copy()
obs["library_id"] = obs.library_id.astype("category")
ad0 = ad.AnnData(obs=obs); ad0.obsm["spatial"] = xy
key = f"knn_k{k}"; t0 = time.perf_counter()
sq.gr.spatial_neighbors_knn(ad0, library_key="library_id",
    n_neighs=k, n_jobs=1, key_added=key)
c = ad0.obsp[f"{key}_connectivities"].tocsr()
d = ad0.obsp[f"{key}_distances"].tocsr()
size = lambda x: x.data.nbytes+x.indices.nbytes+x.indptr.nbytes
r = {"version":sq.__version__,"n":ad0.n_obs,"k":k,
     "library_sizes":m.groupby("library_id").size().to_dict(),"nnz_c":c.nnz,
     "nnz_d":d.nnz,"bytes_c":size(c),"bytes_d":size(d),
     "degree_min":int(np.diff(c.indptr).min()),
     "degree_median":float(np.median(np.diff(c.indptr))),
     "components":int(csgraph.connected_components(c,directed=False,return_labels=False)),
     "seconds":time.perf_counter()-t0}
Path("graph-run").mkdir()
ad0.write_h5ad("graph-run/graph.h5ad")
Path("graph-run/resource.json").write_text(json.dumps(r,indent=2),encoding="utf-8")
print(json.dumps(r))
```

检查每片节点数、degree分布、连通分量和nnz；不硬性要求每行正好k条边，因为实现可能对图对称化，重复坐标也会影响结果。Squidpy把connectivities和distances存入AnnData.obsp，参数写入uns对应key；给每次运行独立key避免覆盖旧图。可将图叠加到空间坐标或组织图像，抽查空洞、边界和跨组织边。

从观测稀疏结构估资源

CSR字节数由data、indices和indptr数组组成，代码分别读取两张图的实际占用。kNN图的边数通常随节点数和k近似线性增长，不应按N×N稠密矩阵估算；但峰值还包括坐标索引结构、临时数组、AnnData元数据、写盘缓存和同机任务。压缩h5ad大小也不能代表运行峰值。

先测代表切片的墙钟时间、进程RSS、nnz和CSR字节，再测全切片与拟用并行度。RSS高但CSR很小时，排查是否重复载入数据或有下游步骤将图转稠密。记录软件版本、坐标单位/变换、library_key、k或半径、n_jobs、各片节点数、nnz和峰值RSS。若实测资源无法满足任务时限，再按实际规模评估计算配置。

不要只因为k越小图越省内存就选最小值。可在代表切片上比较相邻的几个k值，观察连通分量、节点度分布、边长分位数和下游统计是否稳定；这属于方法敏感性检查，不能用单一指标替代实验设计。对组织间隔明显的数据，叠图尤其要检查kNN是否跨越空洞连接到远处区域。

输出核验时逐片对比输入节点数与图矩阵维度，并确认所有观测都属于预期library。将connectivities与distances分别存档，保存建图参数和输入坐标哈希；后续若更改坐标尺度、筛选细胞或邻居数，应使用新key和新结果目录，避免误把旧图当成新结果。

参考资料

Squidpy kNN图接口 官方文档
Squidpy邻域图接口与弃用说明 官方文档