时空组学 聚类分析

时空转录组系列课程

深圳国家基因库 陈静

大纲

背景和意义

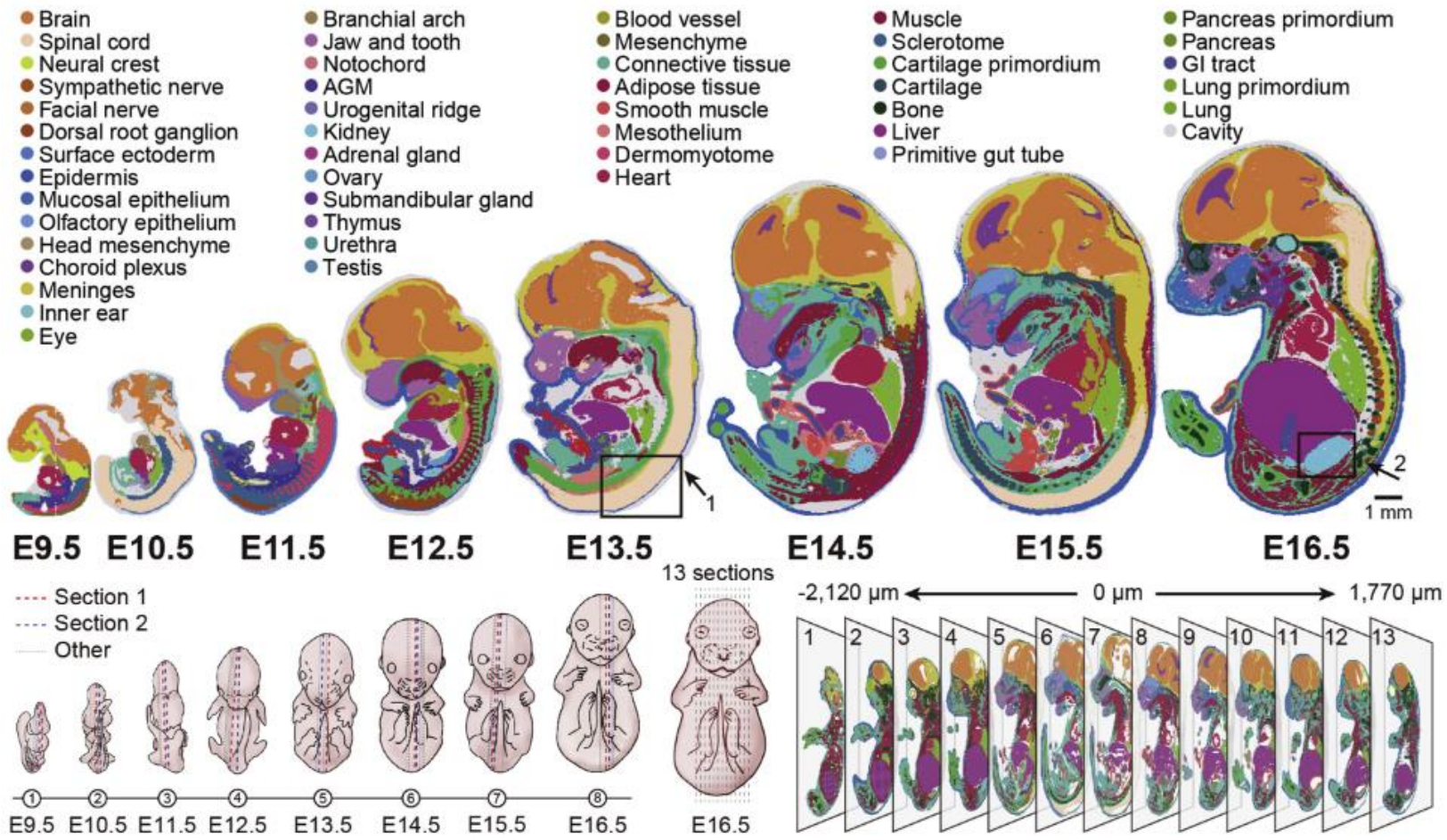
细胞、组织
静态状态

?

时间、空间
动态变化

发育、分化
疾病、治疗

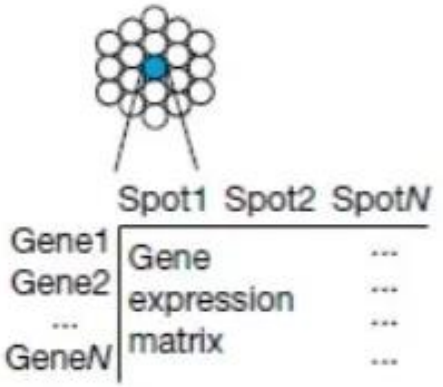
时空组学

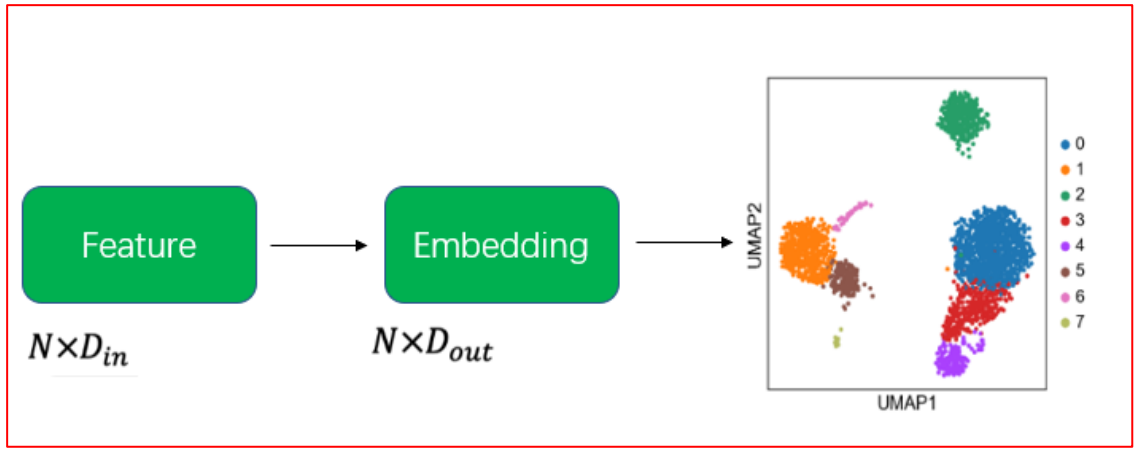


无监督聚类
UMAP可视化

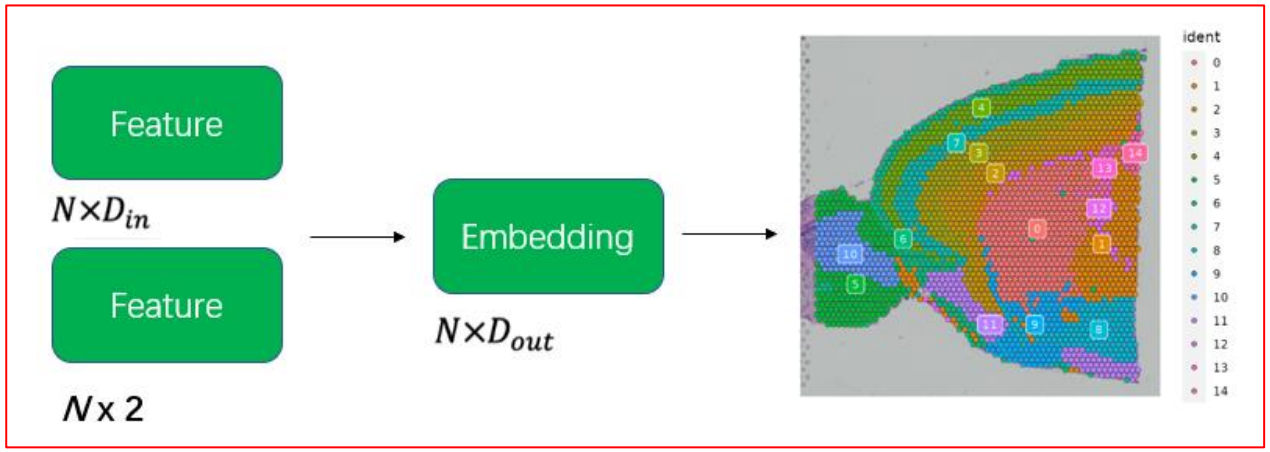
Ao Chen et al. "Spatiotemporal transcriptomic atlas of mouse organogenesis using DNA nanoball-patterned arrays". ***Cell***. 2022 May 12;185(10):1777-1792.e21. https://doi.org/10.1016/j.cell.2022.04.003

- 单细胞：细胞的基因表达信息，研究细胞类型的多样性和功能特征
- 时空组：细胞的基因表达和空间位置信息，研究细胞的功能变化和组织器官的演化过程



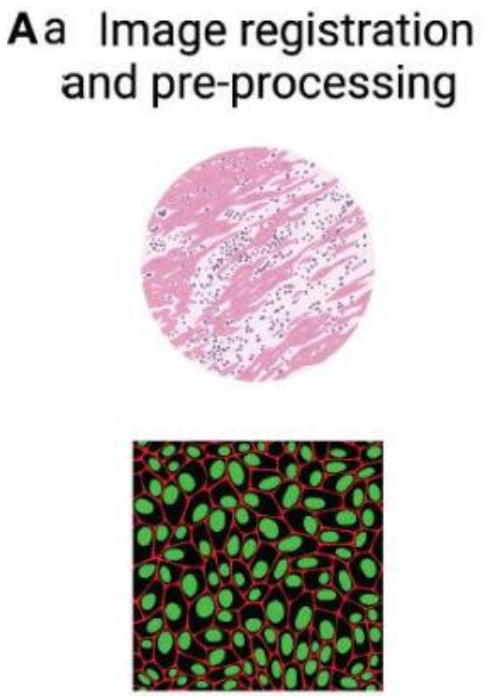


单细胞

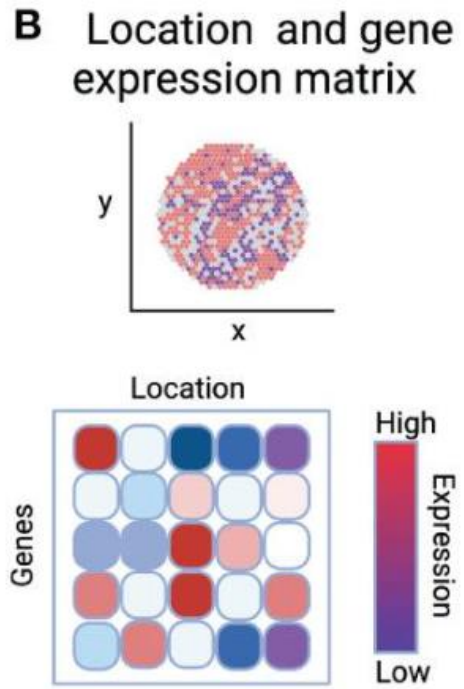


时空组

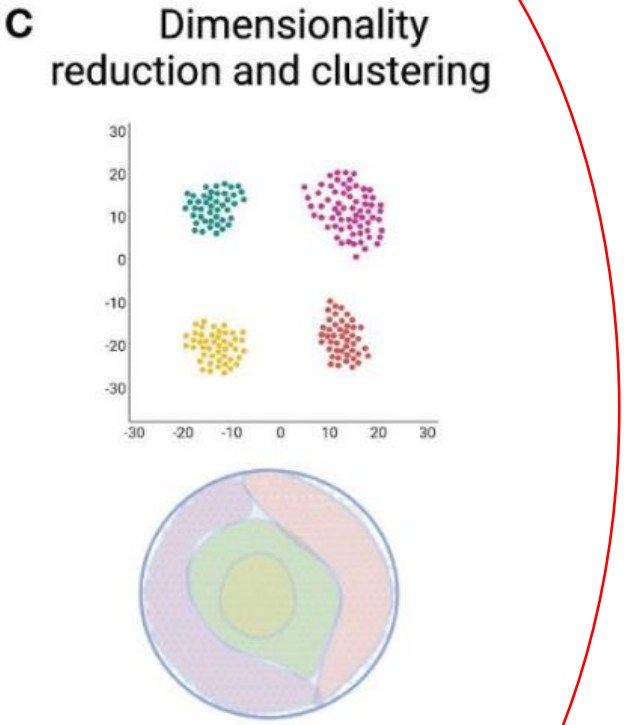
时空组学数据分析关键步骤



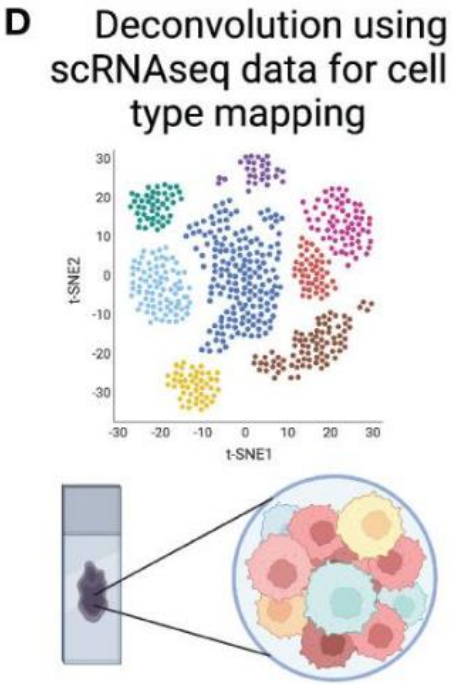
质控、图像配准



计数矩阵、细胞坐标



标准化、降维
聚类



基因表达可视化
组织图像

大纲

数据质控

获取数据

细胞过滤

基因过滤

成年小鼠大脑的 10 μm 冠状切片

1、下载数据：

wget https://cf.10xgenomics.com/samples/spatial-exp/1.1.0/V1_Adult_Mouse_Brain_Coronal_Section_2/V1_Adult_Mouse_Brain_Coronal_Section_2_filtered_feature_bc_matrix.h5
wget https://cf.10xgenomics.com/samples/spatial-exp/1.1.0/V1_Adult_Mouse_Brain_Coronal_Section_2/V1_Adult_Mouse_Brain_Coronal_Section_2_spatial.tar.gz
(https://db.cngb.org/stomics/datasets/STDS0000030)

2、加载数据：

```
├── spatial
│   ├── aligned_fiducials.jpg
│   ├── detected_tissue_image.jpg
│   ├── scalefactors_json.json
│   ├── tissue_hires_image.png
│   ├── tissue_lowres_image.png
│   └── tissue_positions_list.csv
├── V1_Adult_Mouse_Brain_Coronal_Section_2_filtered_feature_bc_matrix.h5
└── V1_Adult_Mouse_Brain_Coronal_Section_2_spatial.tar.gz
```

```python
import scanpy as sc
adata=sc.read_visium(path="./",count_file="V1_Adult_Mouse_Brain_Coronal_Section_2_filtered_feature_bc_matrix.h5")
adata
```

```
AnnData object with n_obs × n_vars = 2807 × 32285
    obs: 'in_tissue', 'array_row', 'array_col'
    var: 'gene_ids', 'feature_types', 'genome'
    uns: 'spatial'
    obsm: 'spatial'
```

数据质控

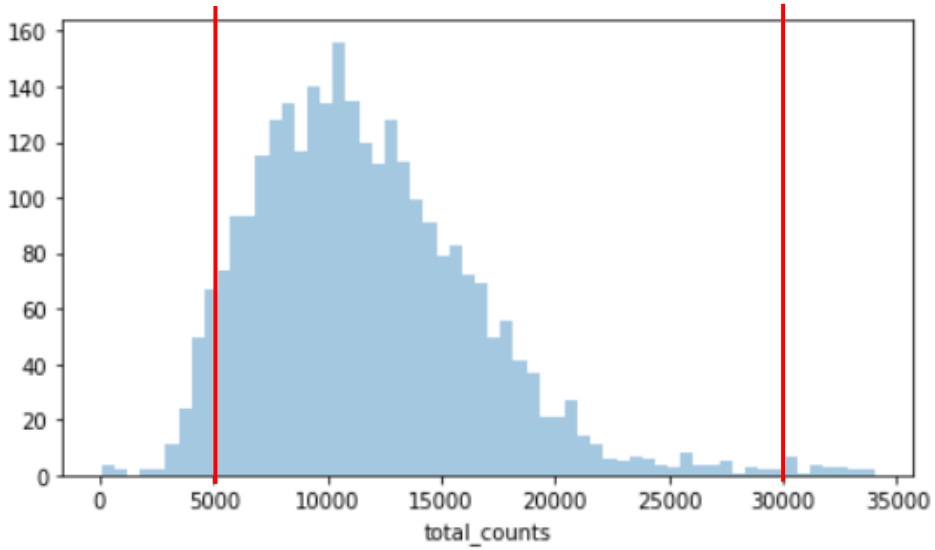
获取数据

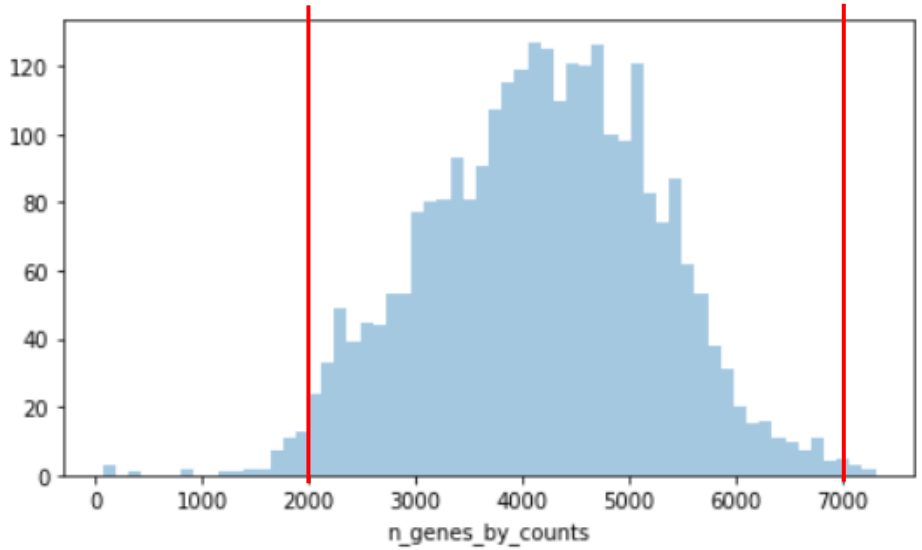
细胞过滤

基因过滤

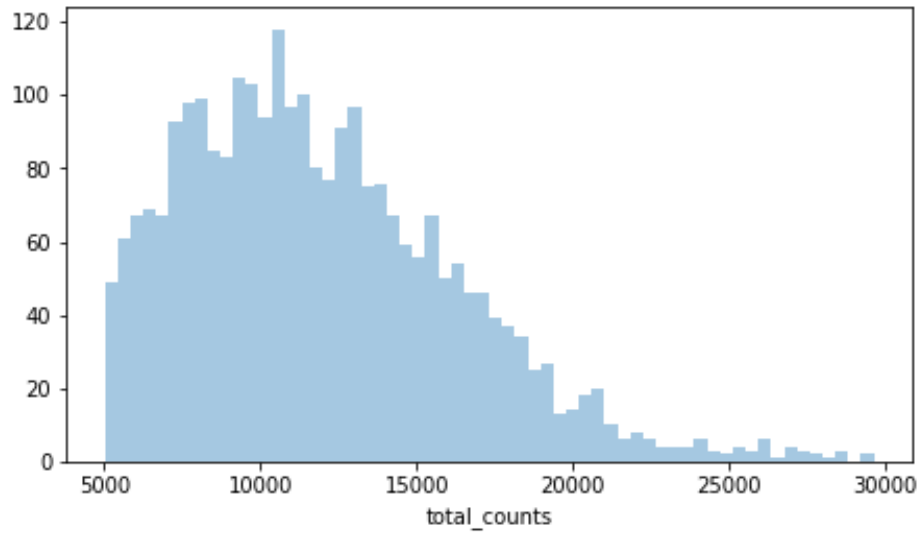
根据细胞总计数和基因表示数过滤低质量的细胞

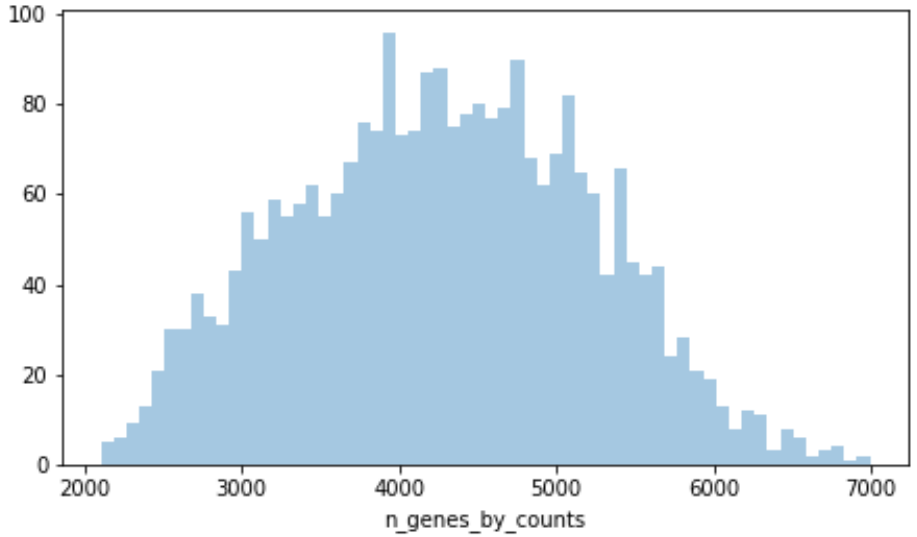
```
sc.pp.filter_cells(adata,min_counts=5000)
sc.pp.filter_cells(adata,max_counts=30000)
sc.pp.filter_cells(adata, min_genes=2000)
sc.pp.filter_cells(adata, max_genes=7000)
```





过滤前





过滤后

数据质控

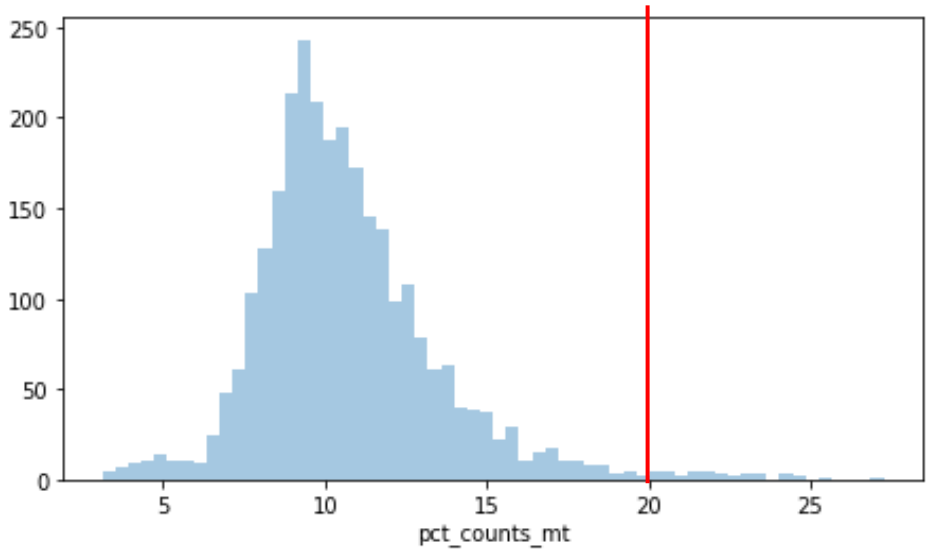
获取数据

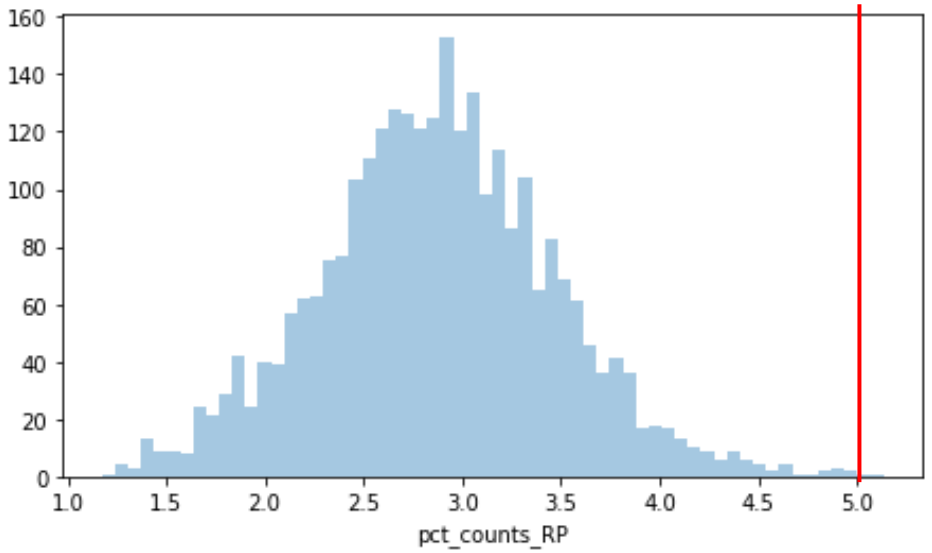
细胞过滤

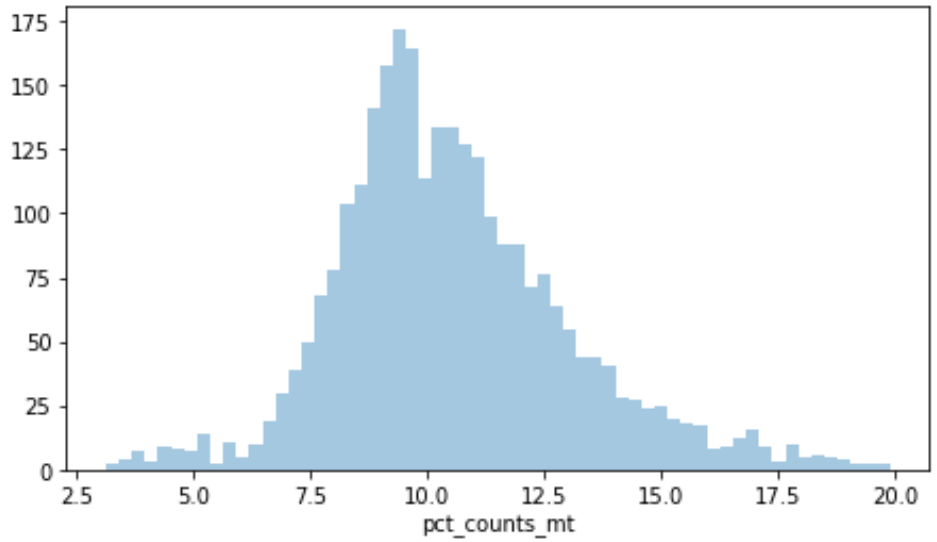
基因过滤

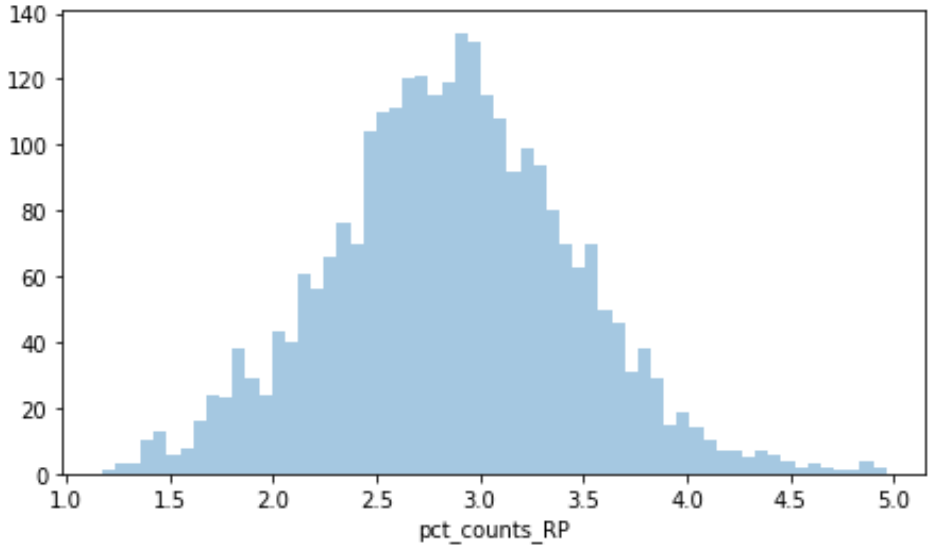
过滤掉线粒体基因和核糖体基因表达高的细胞

```
adata=adata[adata.obs["pct_counts_mt"]<20]
adata=adata[adata.obs["pct_counts_RP"]<5]
```





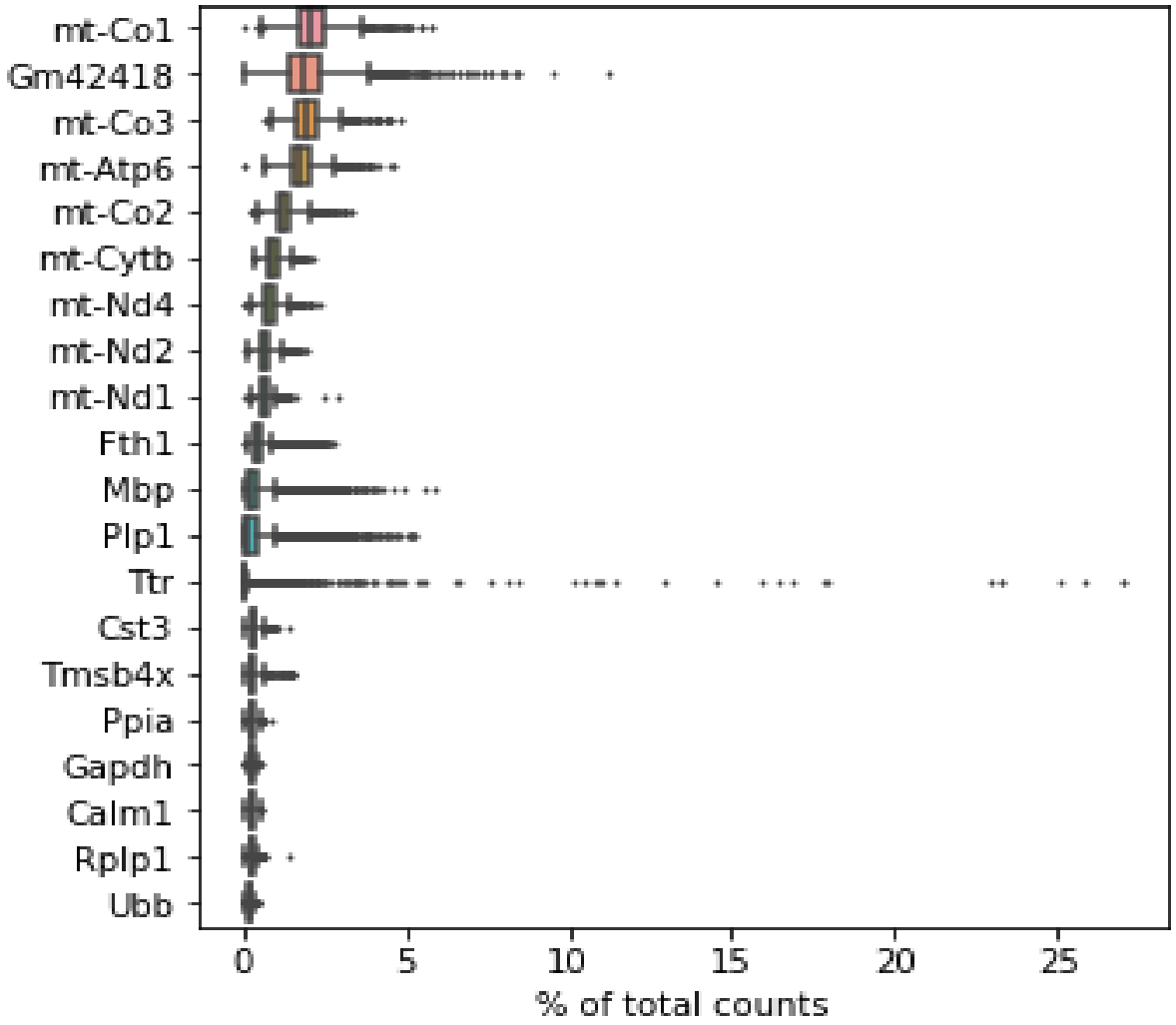




数据质控

根据基因的表达量、基因的检测频率过滤低质量的基因

```
sc.pp.filter_genes(adata, min_counts=1) # 过滤掉在所有细胞中表达计数小于1的基因
sc.pp.filter_genes(adata, min_cells=10) # 过滤掉在少于10个细胞中检测到的基因
sc.pl.highest_expr_genes(adata, n_top=20) # 显示具有最高表达水平的20个基因
```



大纲

数据预处理

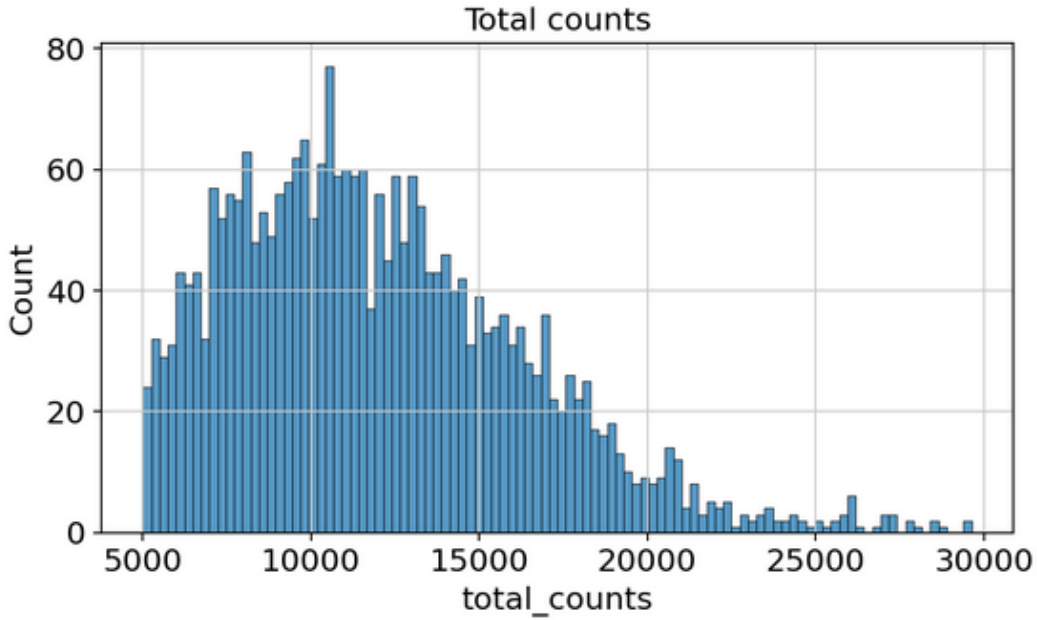
标准化

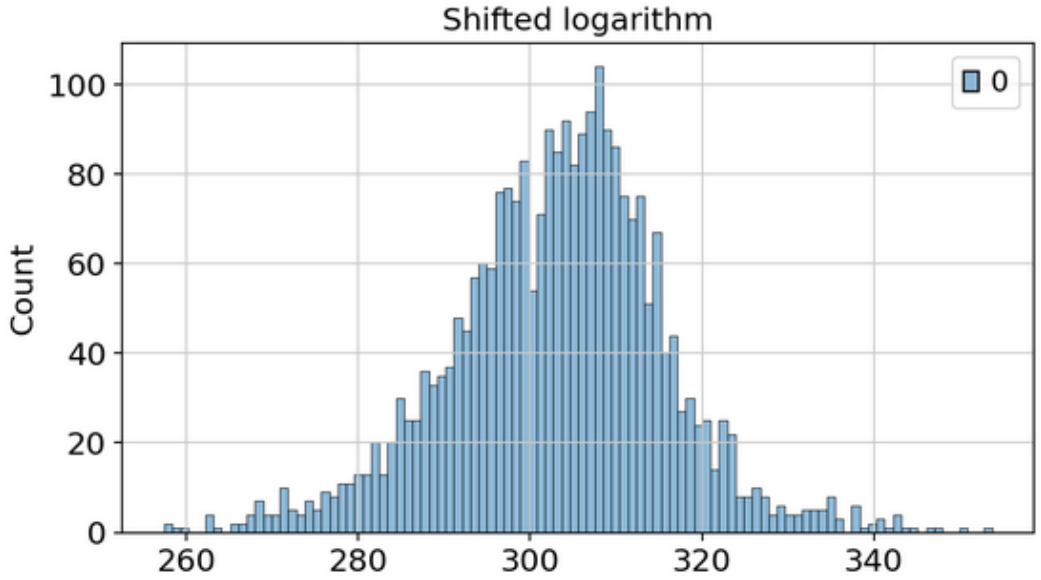
高可变基因

归一化：每个细胞的**测序深度**和**基因长度**不同，导致单个细胞里每个基因表达不在一个水平上，使用CPM等方法对表达矩阵进行标准化处理，使每个细胞里每个基因的表达落在一个可比的水平

取对数：处理偏态分布的数据，使数据更加平滑和可比较，符合**正态分布**的假设

```
sc.pp.normalize_total(adata, inplace=True) # 归一化
sc.pp.log1p(adata) # 取对数
```





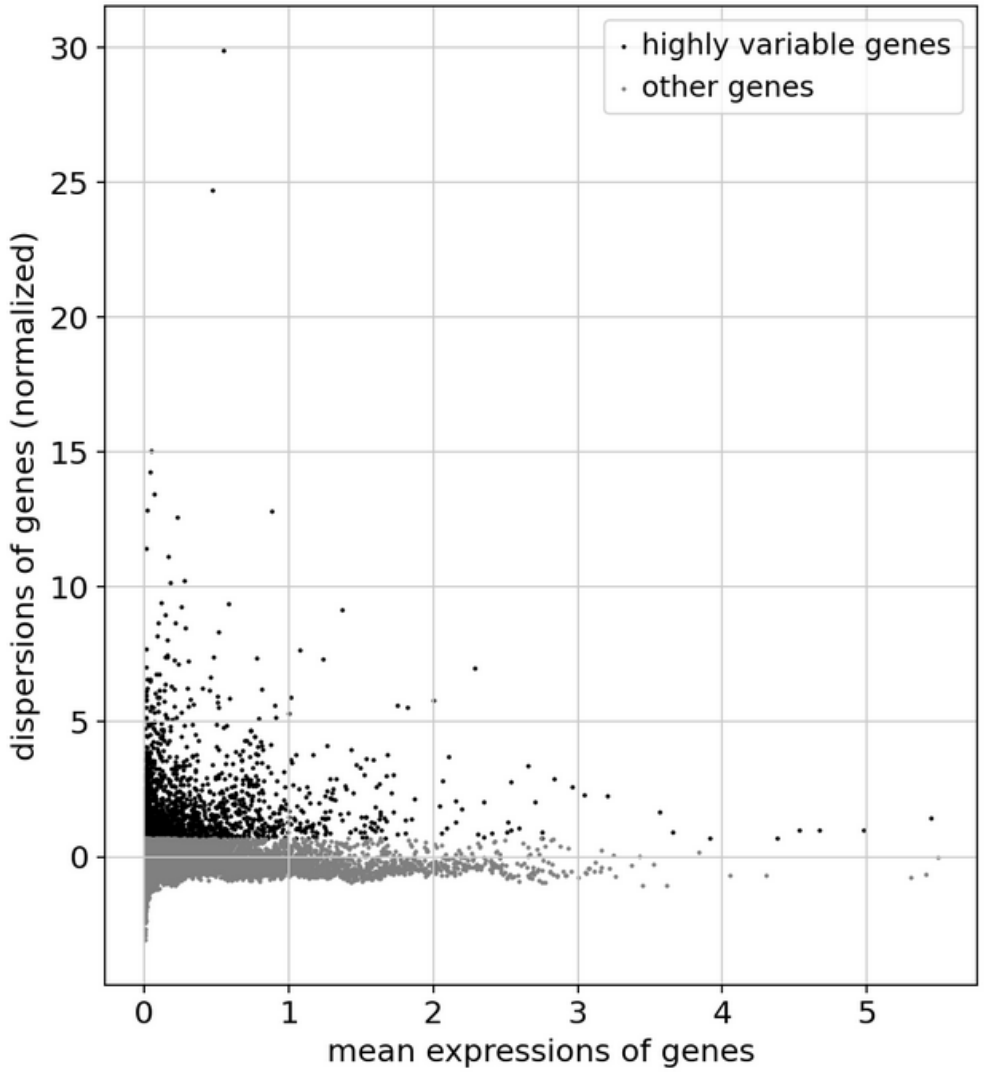
数据标准化的总计数分布近似正态分布

数据预处理

标准化

高可变基因

高可变基因(HVG)：也叫特征基因，在细胞或组织中**表达水平变异较大的基因**，基于基因离散度的方法，去掉大部分不具有分析意义的基因，通常选择1000到5000个HVG用于下游分析




```
sc.pp.highly_variable_genes(adata, flavor="seurat", n_top_genes=2000)
adata=adata[:,adata.var.highly_variable]
```

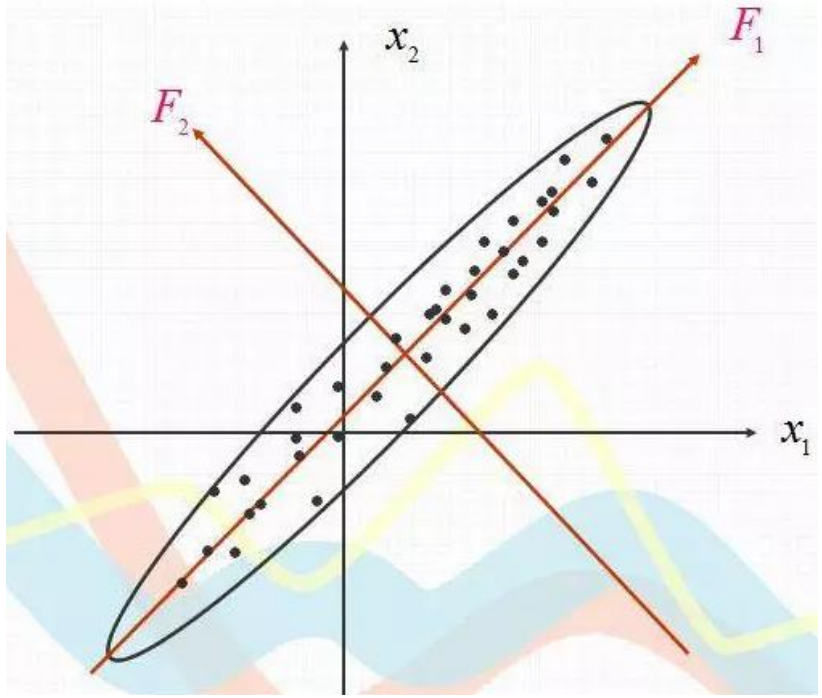
大纲

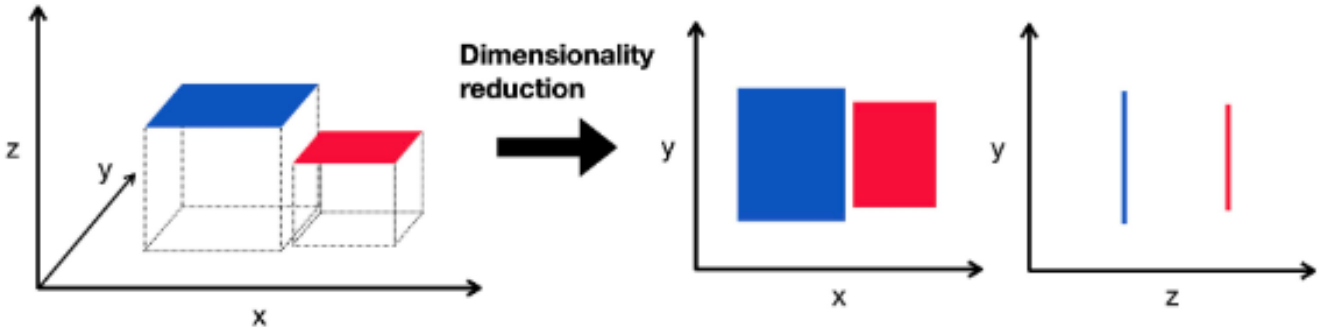
降维

线性降维

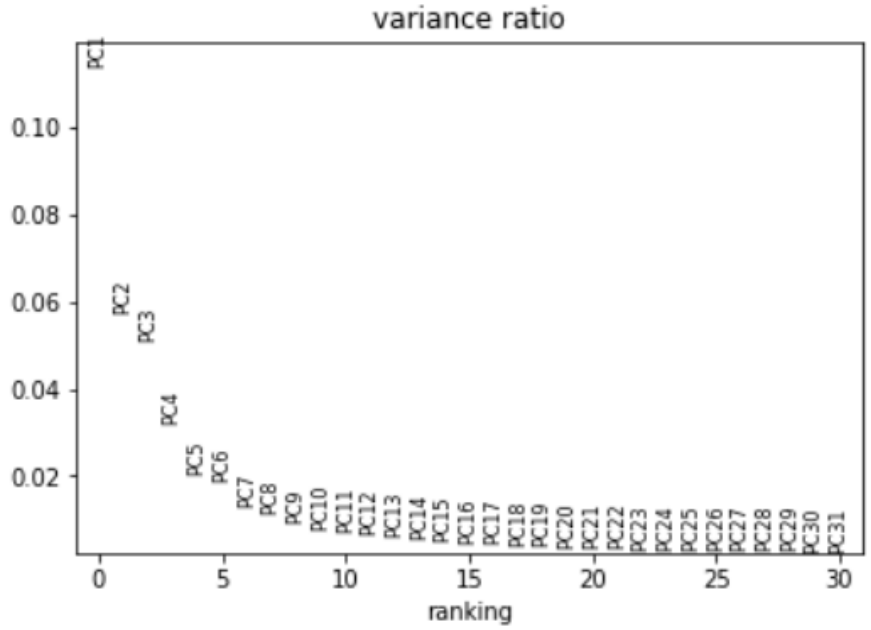
非线性降维

主成分分析(PCA)：线性降维方法，用于将高维数据转换为低维空间，保留主要信息，有助于可视化和理解数据，同时减少计算复杂性





sc.pp.pca(adata)

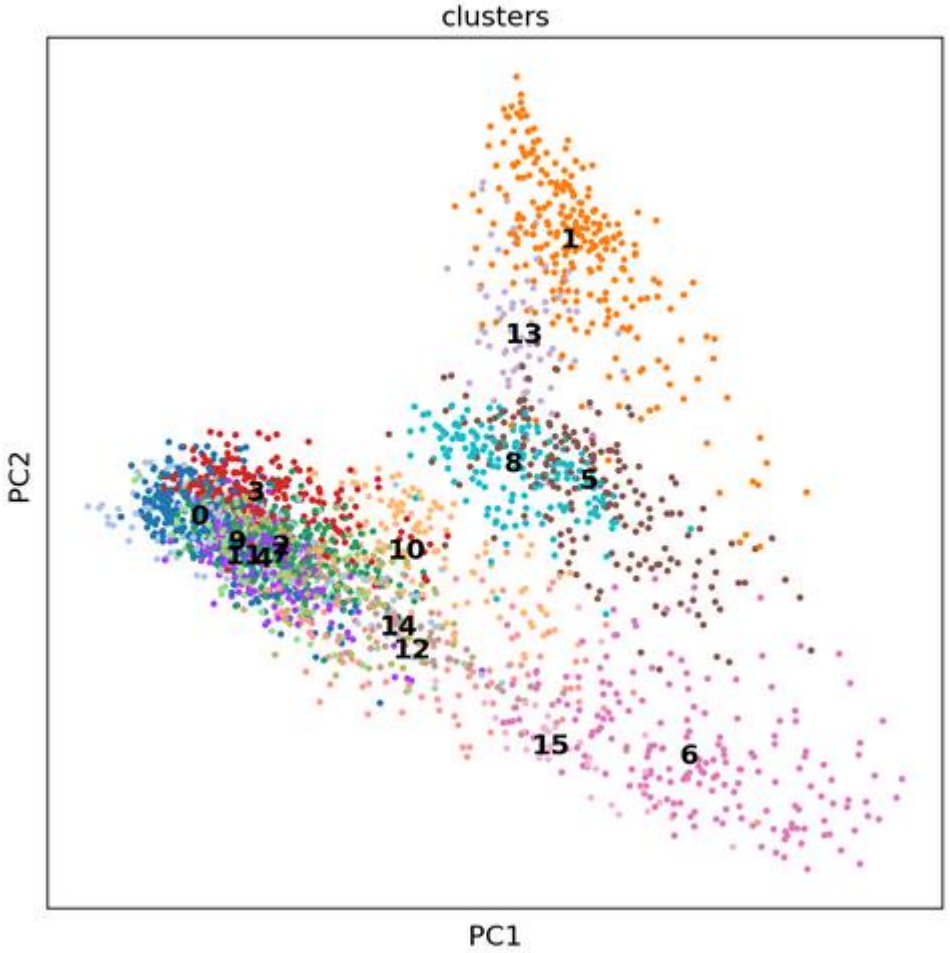


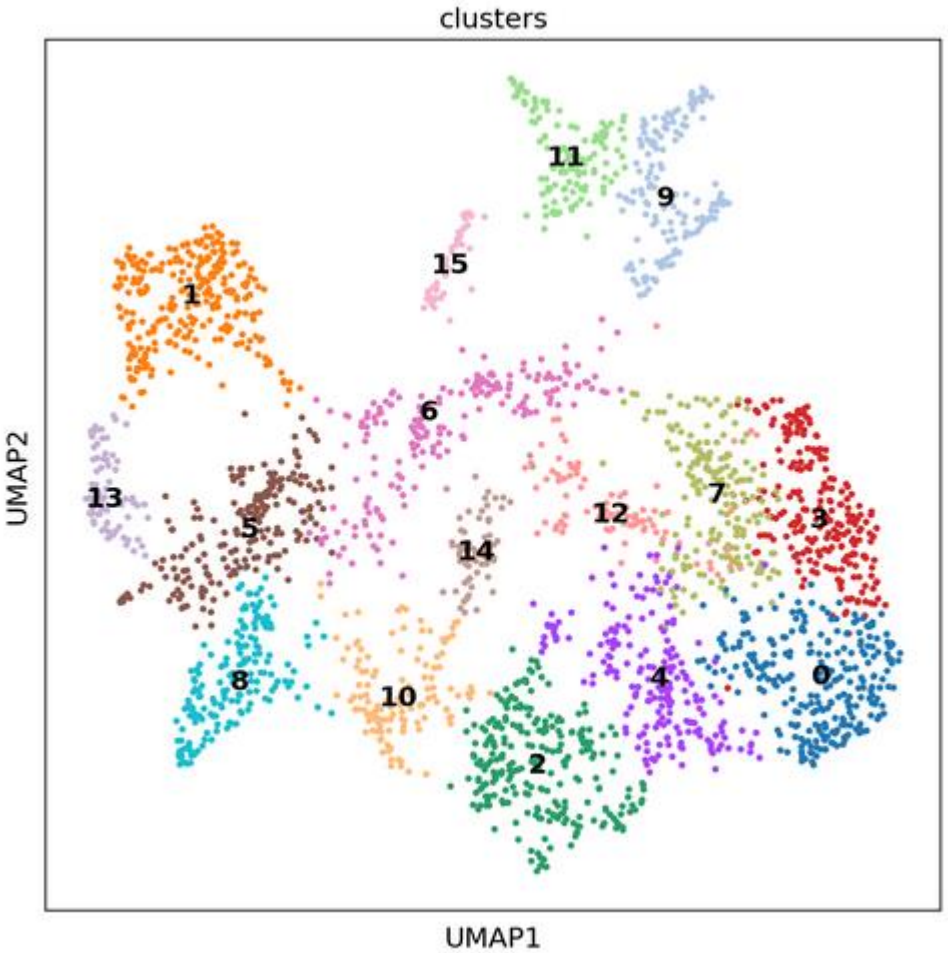
降维

线性降维

非线性降维

PCA降维的局限性：无法捕捉到非线性结构数据的特征；而且虽然保留了大部分数据的方差，仍然有信息丢失的风险。



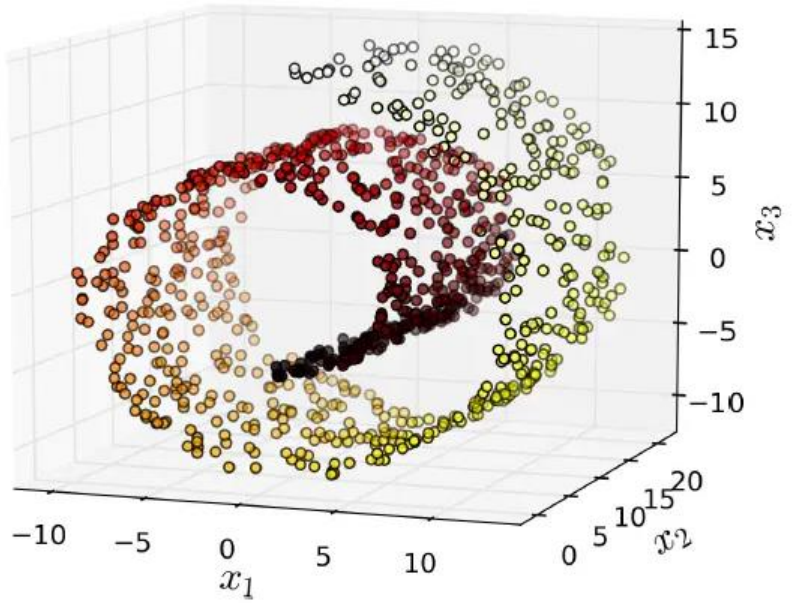


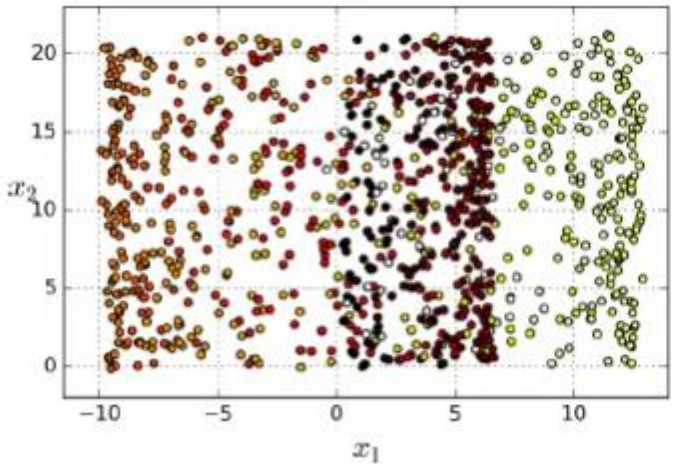
降维

线性降维

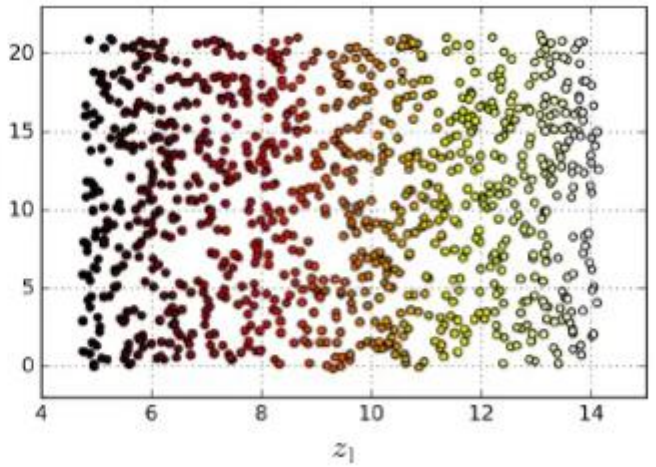
非线性降维

基于流形学非线性降维: 如下面经典的瑞士卷模型， 嵌入三维空间的二维流形，卷曲的操作并没有产生关于原始结构的新信息





投影



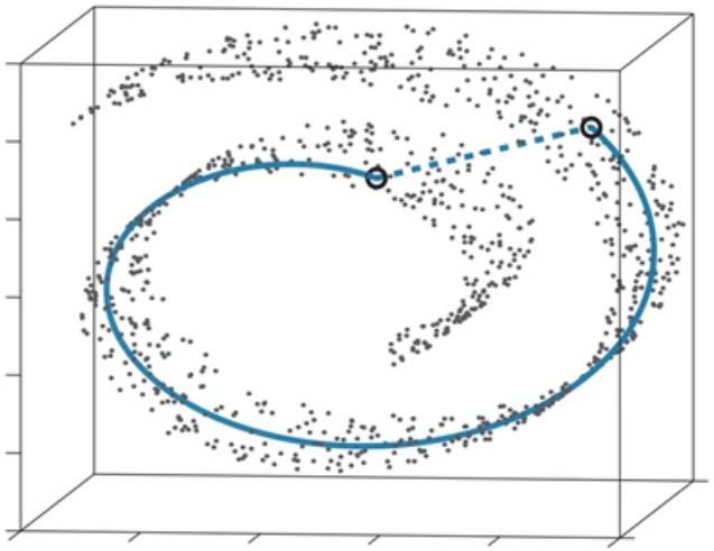
预想

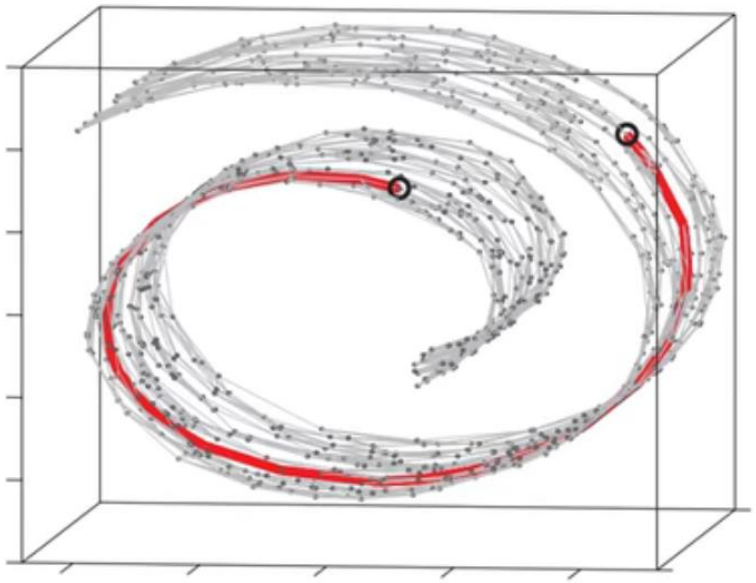
降维

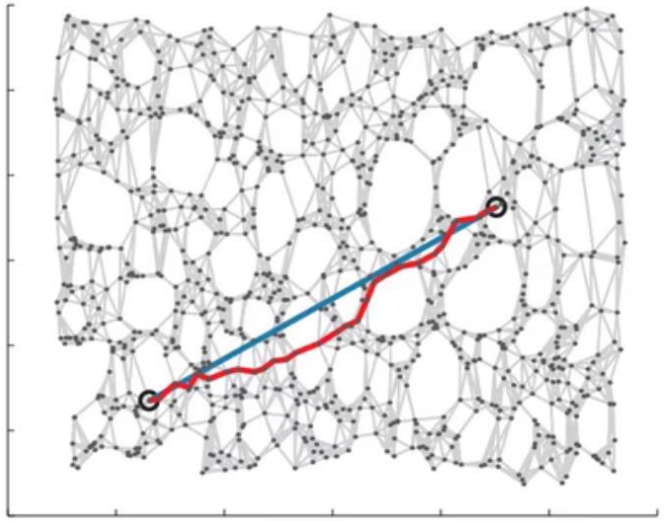
线性降维

非线性降维

非线性降维: 确定各个点之间的位置关系，构造高维的数据分布结构，投影到低维空间，然后根据高维空间点与点之间的相对关系，提取特征值，在低维空间中重构这种距离关系







Dijkstra

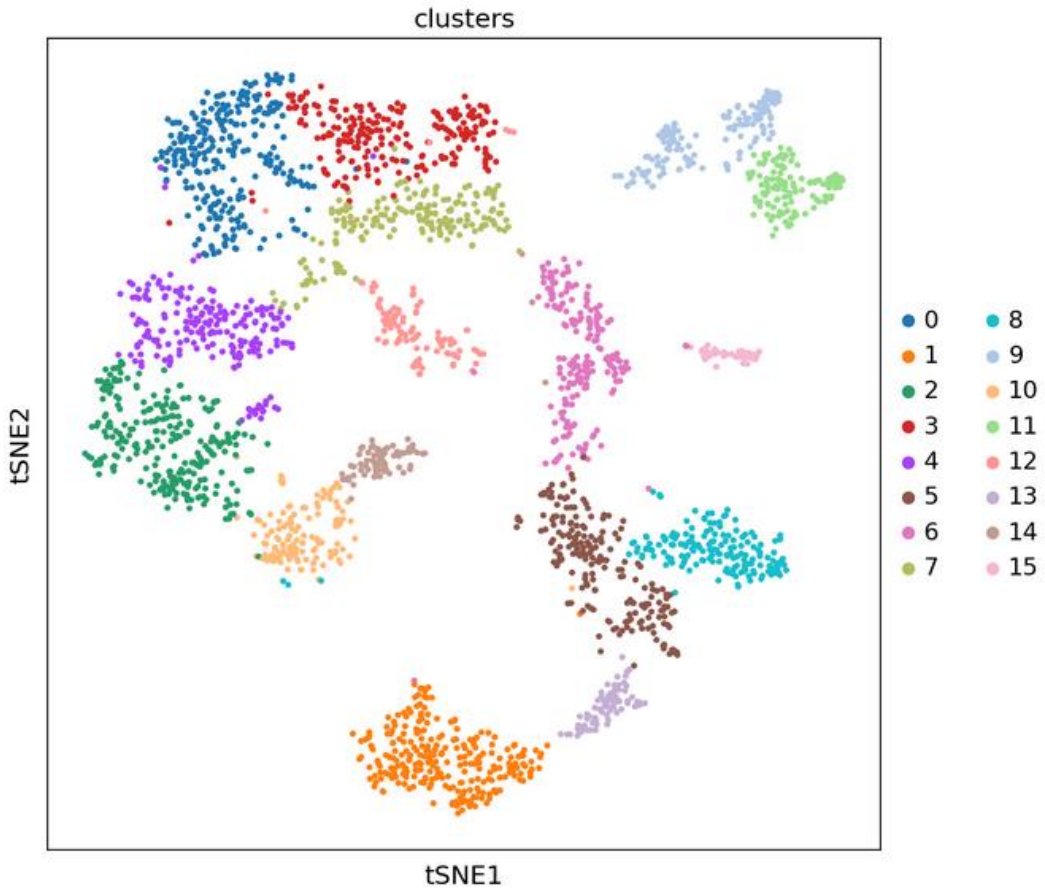
降维

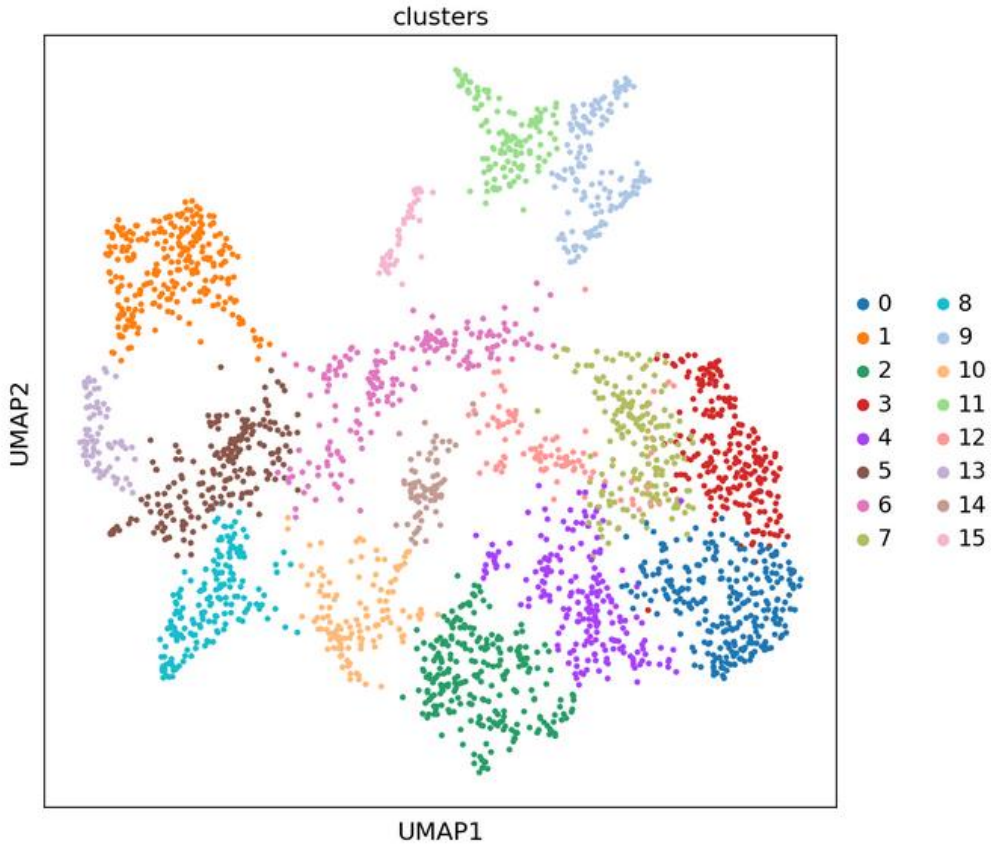
线性降维

非线性降维

tSEN: 注重局部结构，簇间的相似性高，适合亚群内再挖掘的结果展示

UMAP：注重全局结构，最大程度的保留原始数据的特征，运行数据快，适合数据量大的样本





大纲

聚类分析

算法	原理
Leiden	基于图论和社区发现的聚类算法；通过优化模块度来划分细胞的聚类，以发现具有相似基因表达模式的细胞群。
CCST	基于神经网络无监督聚类算法；将细胞根据其基因表达模式划分为不同的细胞周期状态，然后将相似的细胞周期状态聚类在一起。
BayesSpace	基于概率图模型的聚类算法；通过建立贝叶斯模型来估计每个细胞或样本在空间上的概率分布，将细胞或样本按照其转录组数据的空间分布进行聚类。
SpaGCN	基于图卷积网络的聚类算法；通过图卷积，从其相邻spot聚合每个spot的基因表达,从而识别具有一致表达和组织学的空间域
GraphST	基于图自我监督对比学习的聚类算法，利用空间信息和基因表达谱进行空间信息聚类、整合和细胞类型去卷积。

聚类分析

聚类原理

Louvain算法

Leiden算法

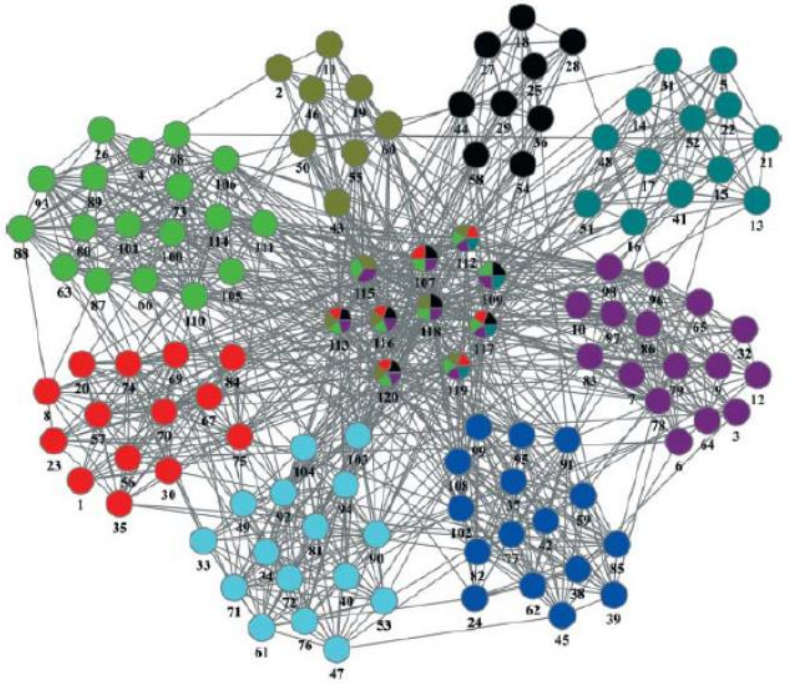
聚类结果

Marker基因

社区发现算法：一种用于在复杂网络中识别出具有紧密连接的节点群体或社区的算法

假设我们有一个图，在这个图中的节点是某个镇的所有的人，在这个图中我们需要确定哪些人之间是一个团体(同一个家庭或同一个小区)

社区发现



聚类分析

聚类原理

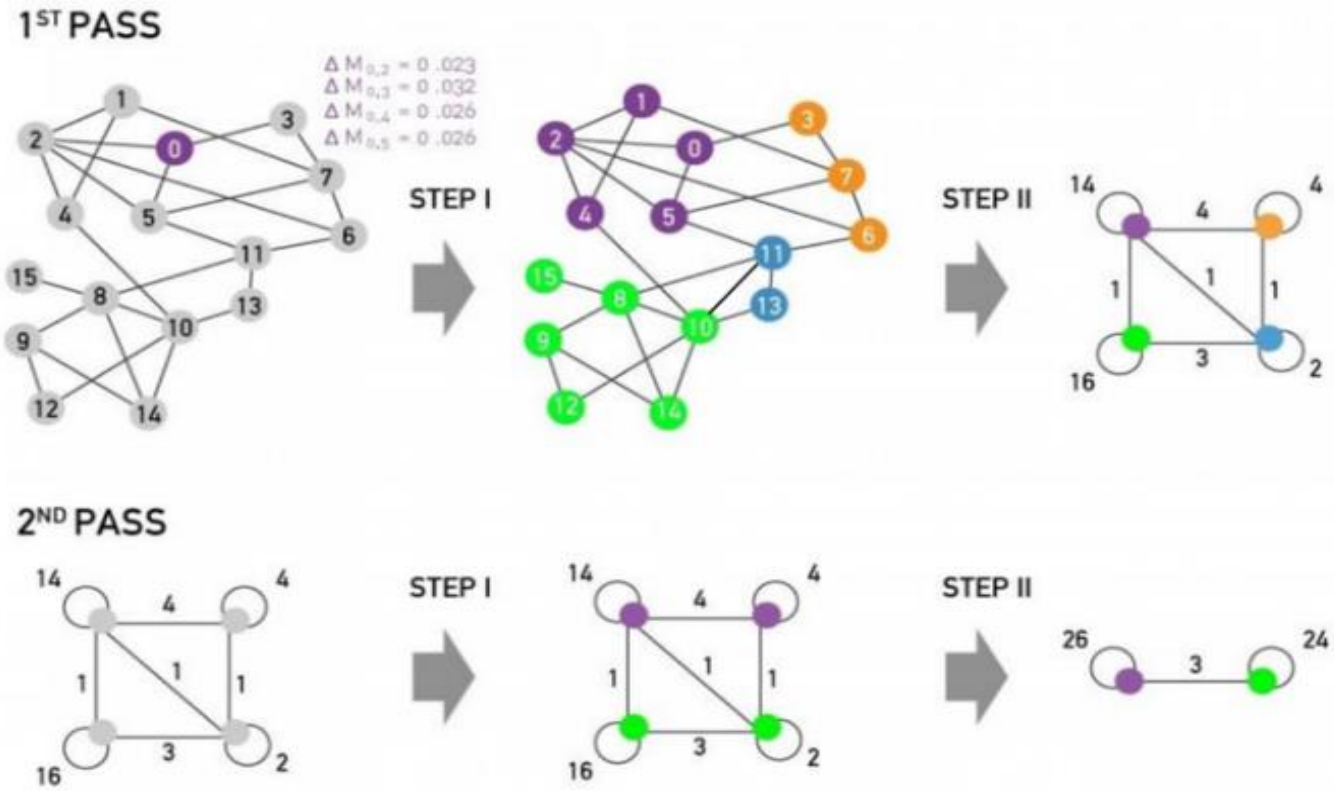
Louvain算法

Leiden算法

聚类结果

Marker基因

- **构建邻接图**：每个点表示为一个节点，相似性计算形成边和权重
- **初始化社区**：每个点初始化为一个单独的社区，计算此时的模块度
- **迭代优化**：合并节点i和节点j为一个新的社区，计算模块度增量，将i节点划分到使模块度增量最大且大于0的那个节点中去，直到验证模块度没有显著增加。
- **社区聚集**：将划分出来的社区聚合为一个巨型节点，重复迭代，知道达到终止条件



——模块度增量

$$\triangle Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

$$\triangle Q = k_{i,in} - \frac{\sum_{tot} \times k_i}{m}$$

【其中，$\sum_{in}$是在社区 c 内的边数；$\sum_{tot}$是在社区 c 内的节点的度数；k_i是节点 i 的度数；$k_{i,in}$是节点 i 和社区 c 内节点之间的连接数和；】

聚类分析

聚类原理

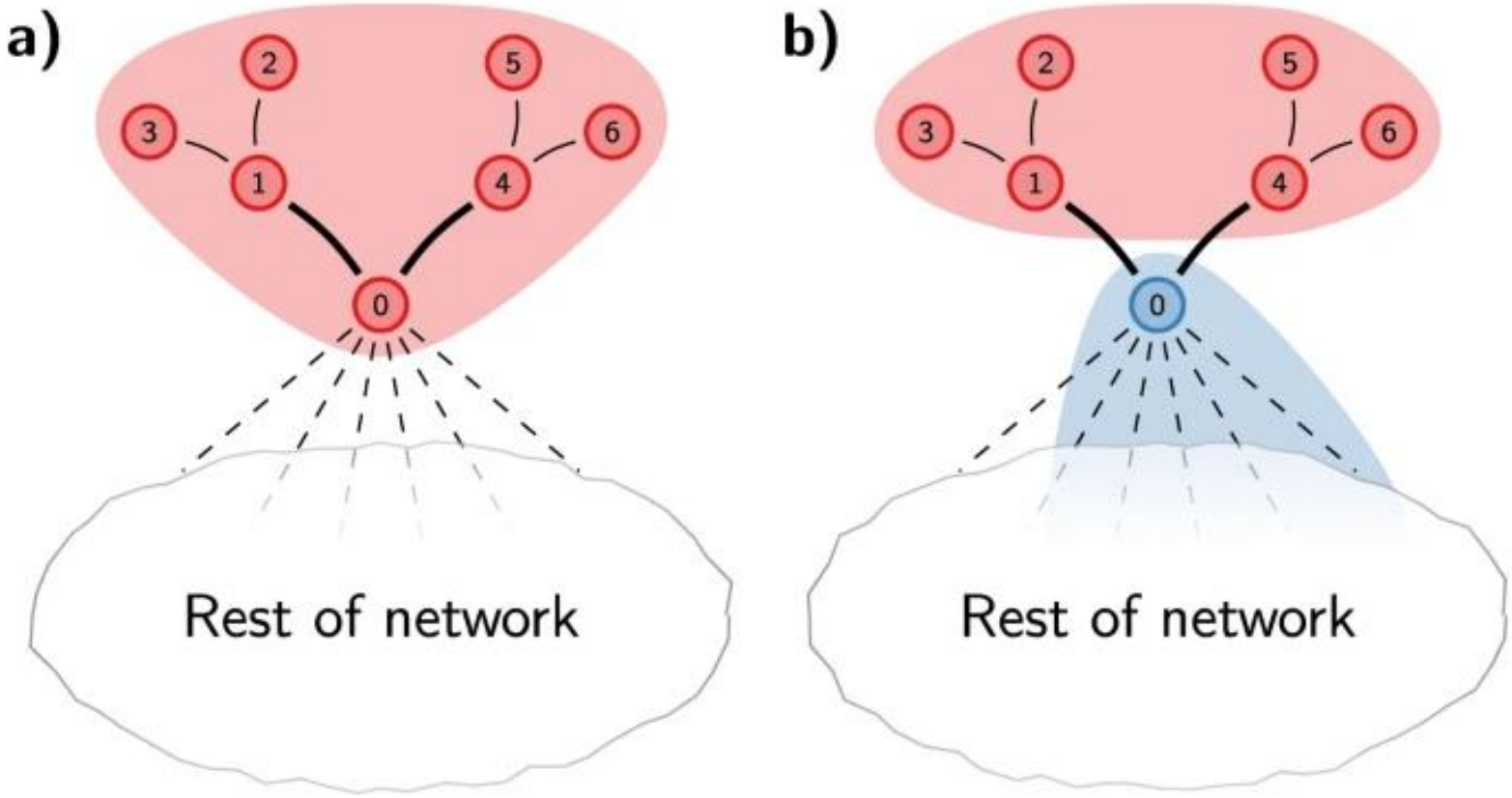
Louvain算法

Leiden算法

聚类结果

Marker基因

Louvain的局限性：社区脱节，下一次迭代的开始可能会把上一轮迭代中的节点移动到其它社区，使得社区内部的连接性差 (0移动到另一个社区，红色区域连接没有断开，分裂为两个社区更好)



聚类分析

聚类原理

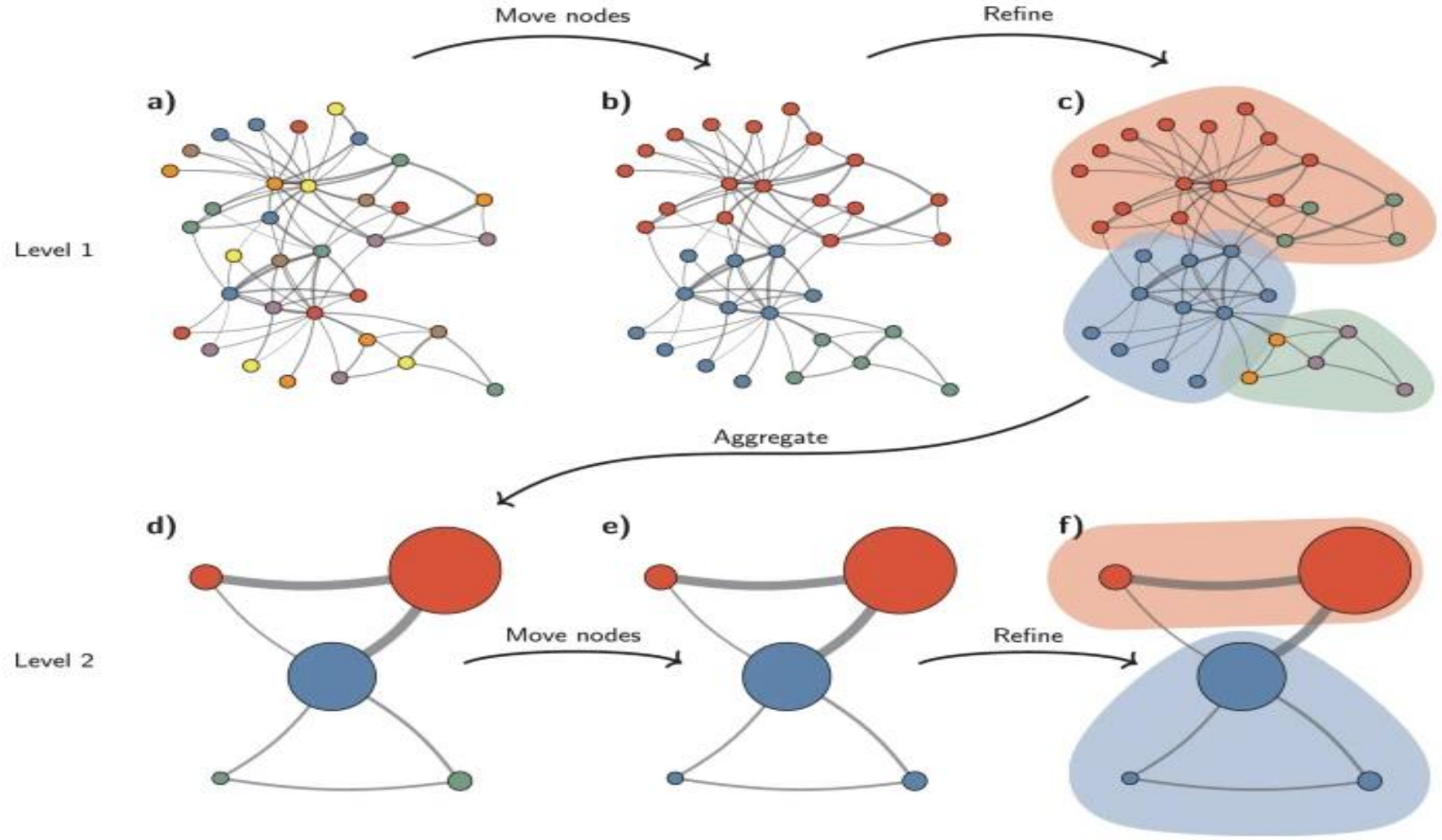
Louvain算法

Leiden算法

聚类结果

Marker基因

a)每个节点视为一个单独的社区; b)移动节点，划分社区；c) 每个社区内部移动节点，Refine改善，**局部优化**(例如红色社区划分2个亚社区)；d)社区形成新的节点；e)移动节点，划分社区；f)模块度不再增加，迭代终止



聚类分析

聚类原理

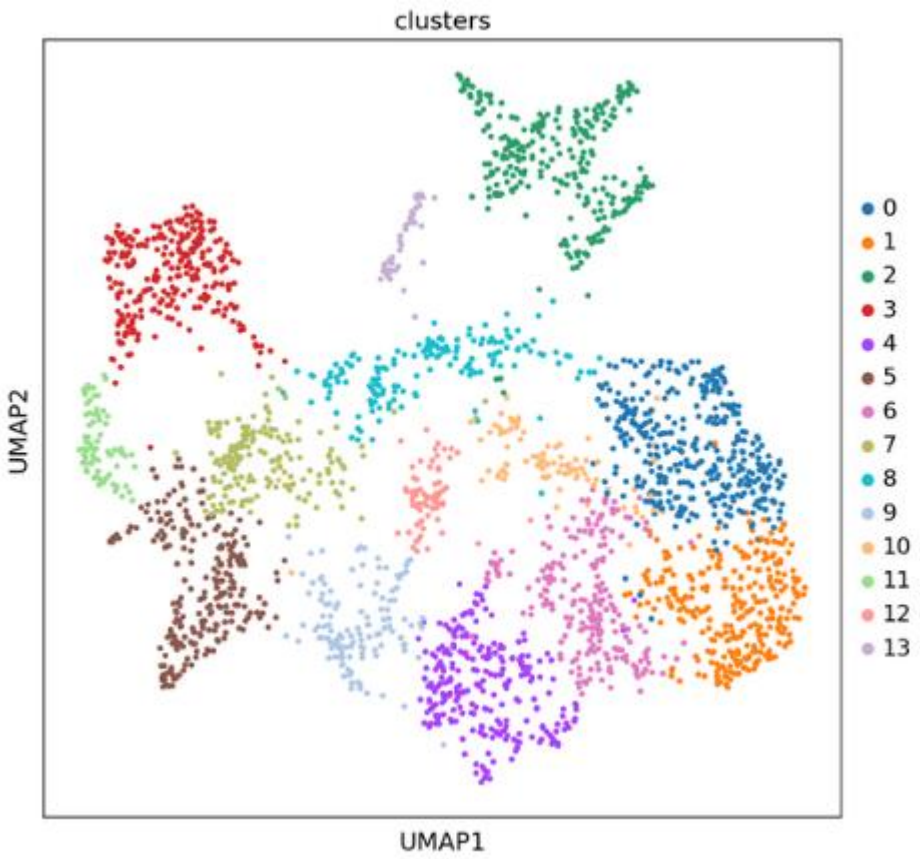
Louvain算法

Leiden算法

聚类结果

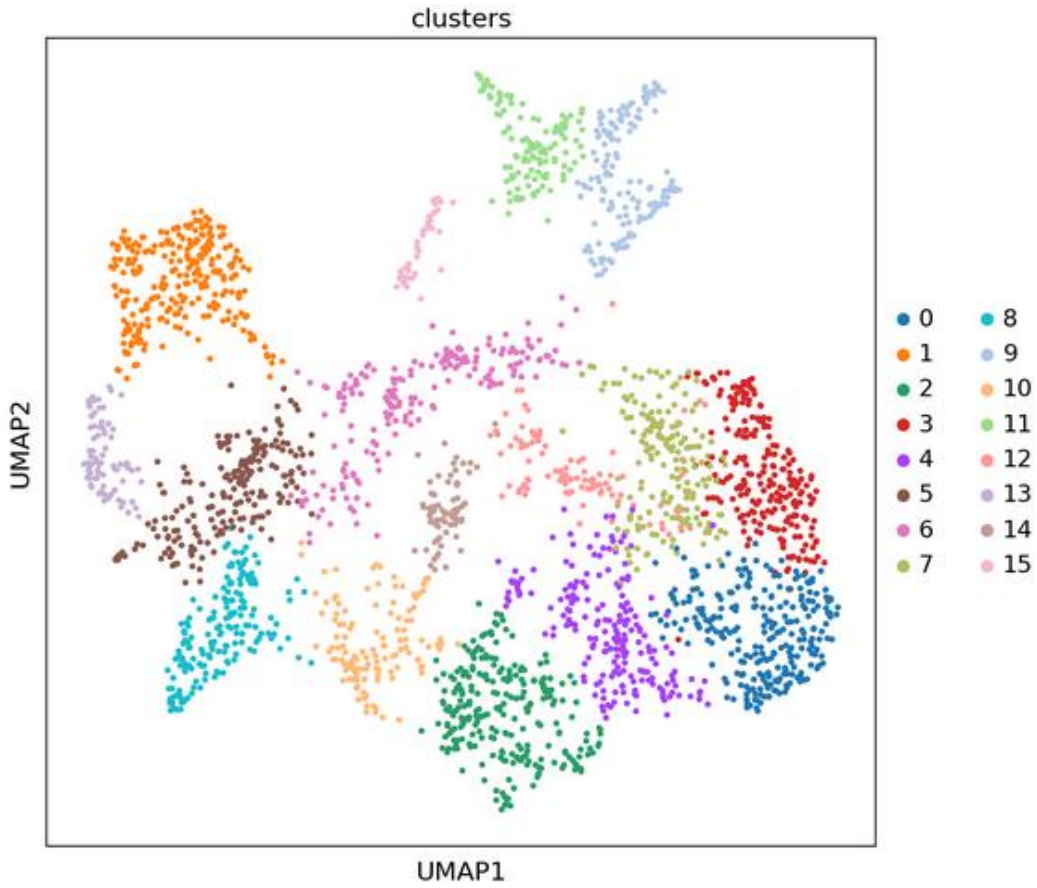
Marker基因

使用Louvain算法聚类



sc.tl.louvain(adata)

使用Leiden算法聚类



sc.tl.leiden(adata)

聚类分析

聚类原理

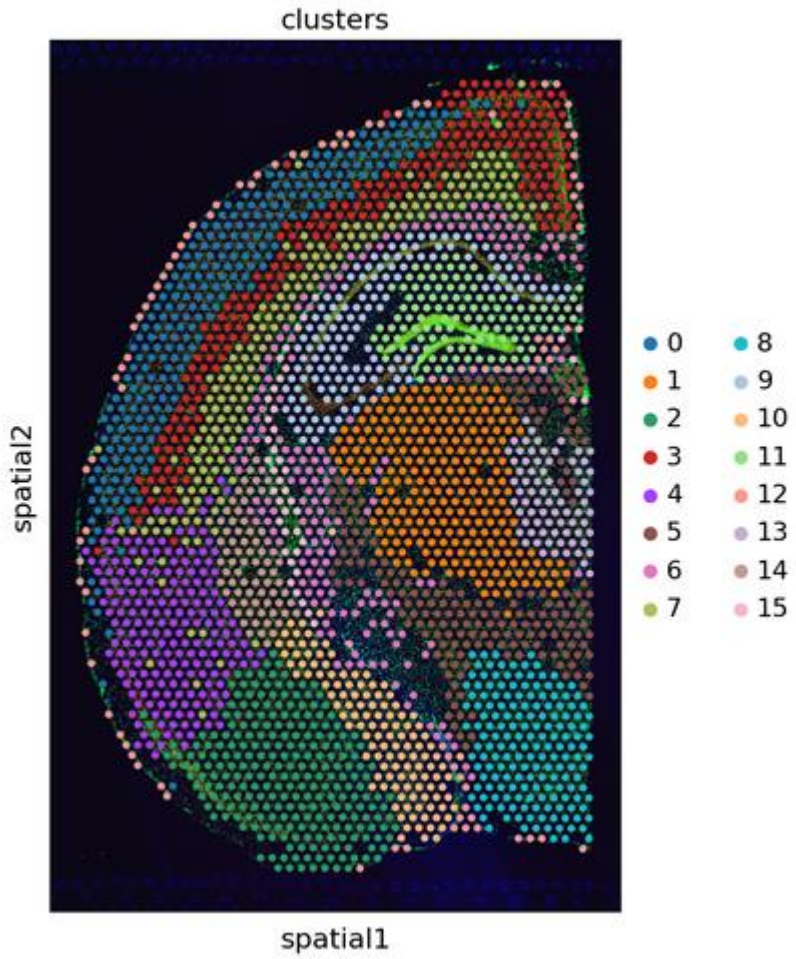
Louvain算法

Leiden算法

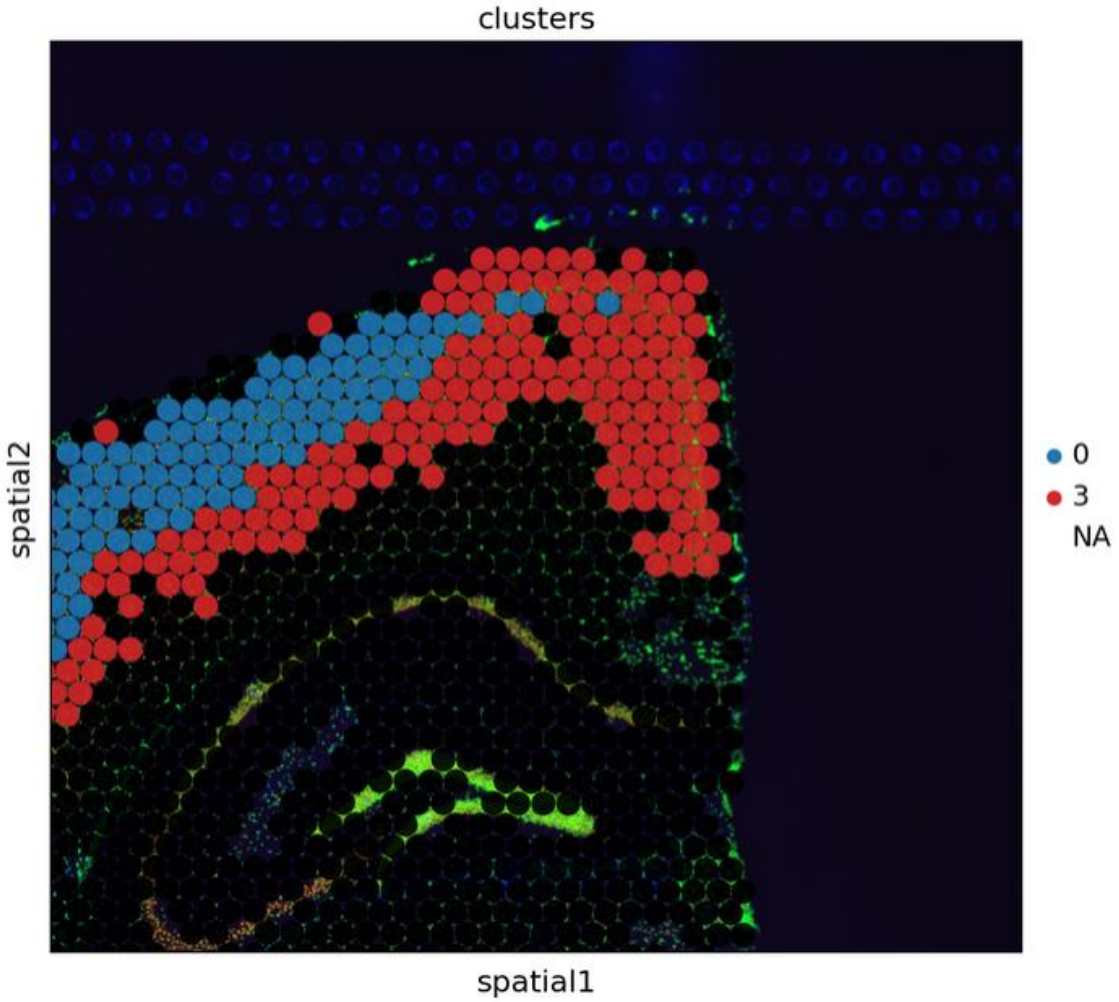
聚类结果

Marker基因

可视化细胞类型在组织中的空间分布



观察指定类别在特定区域的底层组织形态



Marker基因

聚类原理

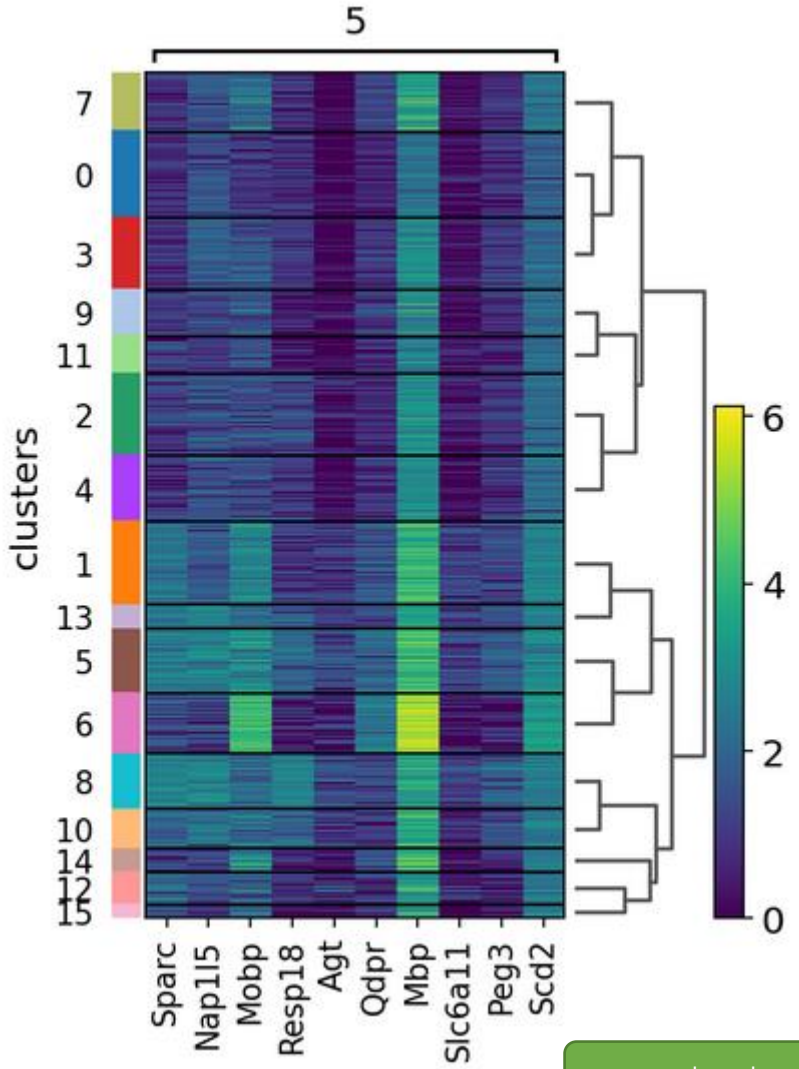
Louvain算法

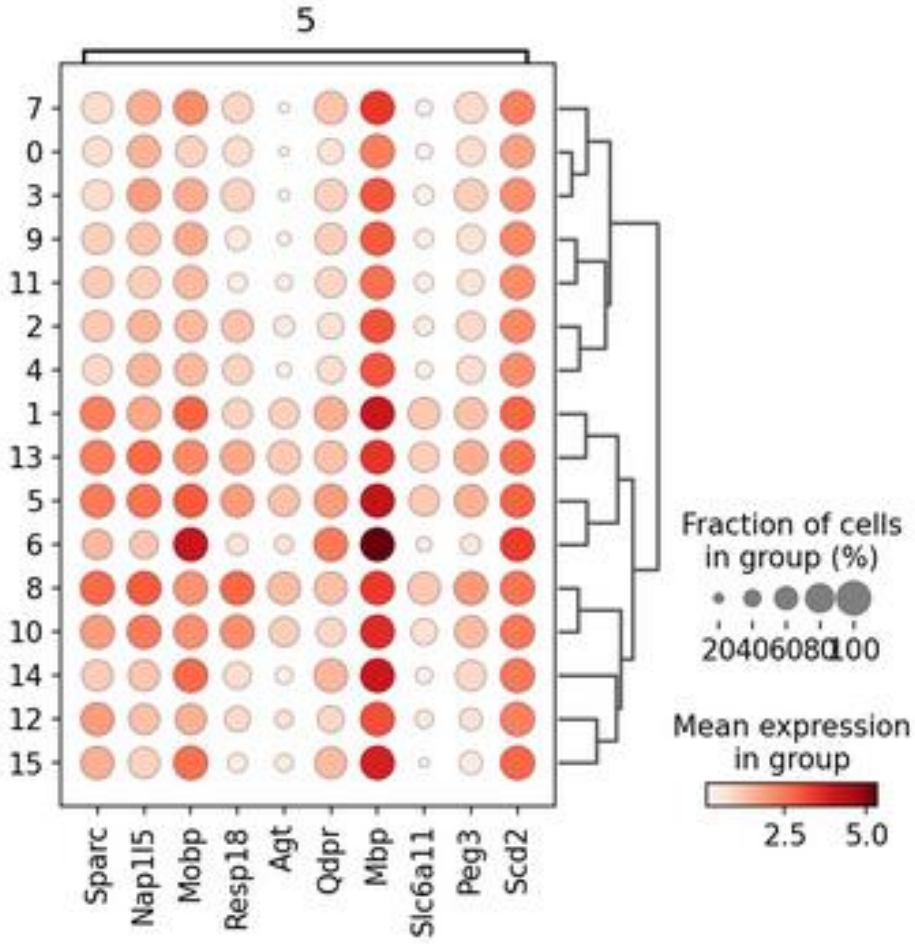
Leiden算法

聚类结果

Marker基因

在特定细胞类型中具有**高度特异性的表达基因**，也叫差异表达基因，常用统计学方法包括t检验、Wilcoxon秩和检验等





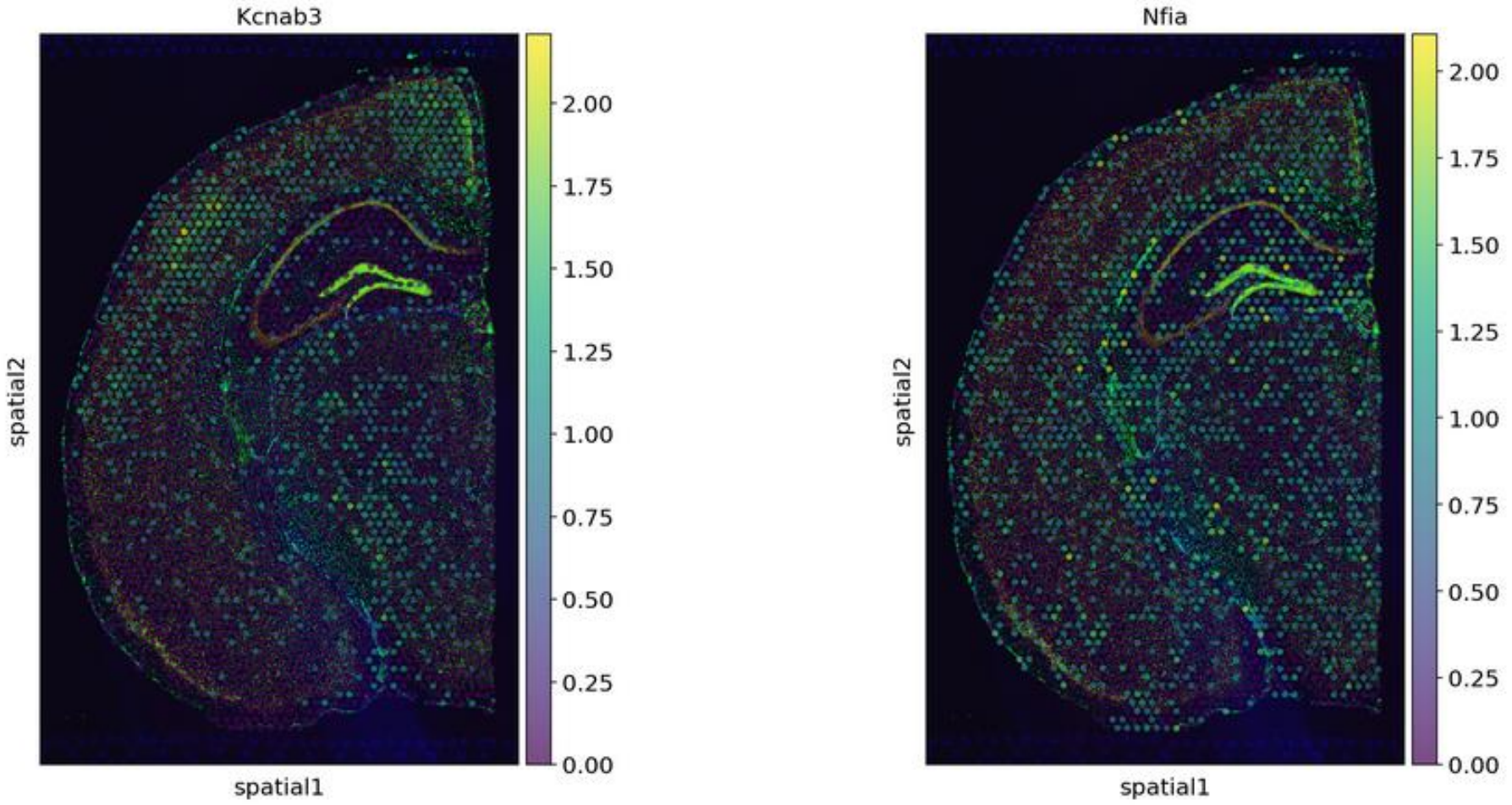

```
sc.tl.rank_genes_groups(adata, "clusters", method="t-test")
```

大纲

空间高可变基因

空间高可变基因: 在组织或器官中表达具有空间分布差异的基因; 提供了关于细胞功能和组织结构变化的重要线索.

SpatialDE: 基于贝叶斯统计模型，通过建立高斯过程模型来描述基因表达在空间上的变化



空间上表达存在显著差异的2个基因

大纲

1. 背景和意义
2. 数据质控
3. 数据预处理
4. 降维
5. 聚类分析
6. 空间高可变基因
7. **代码示例**

代码示例

鹏城AI靶场 https://datai.pcl.ac.cn/

注册 -> 登录 -> 实验 -> 创建实验 ->启动环境(环境镜像选择Python生物分析专用镜像CPU)

在文件里，默认启动了一个Jupyter Notebook，可直接在这里操作

```
#获取数据
Import os
import wfio
wfio.download_from_cloud_disk('path://29fd78f47ae84c5cb6f803d6f86b4334', './Adult_Mouse_Brain_Coronal_Section_filtered_feature_bc_matrix.h5') #下载云盘分享的10X基因表达数据
wfio.download_from_cloud_disk('path://554aa7ea77b7427d85be2fcf8436de51', './V1_Adult_Mouse_Brain_Coronal_Section_2_spatial.tar.gz') #下载云盘分享的组织切片的图像
os.system("tar zxvf V1_Adult_Mouse_Brain_Coronal_Section_2_spatial.tar.gz") #解压，生成spatial目录
```

代码示例

```
#导入依赖库、读取数据
import scanpy as sc #导入Scanpy库并将其命名为sc
import matplotlib.pyplot as plt #导入基本作图库
import seaborn as sns #导入高级作图库
sc.set_figure_params(facecolor="white", figsize=(8, 8)) #设置绘图的背景色为白色,绘图的尺寸为8x8英寸
sc.settings.verbosity = 3 #设置输出信息的详细程度,值为3表示输出最详细的信息
adata=sc.read_visium(path="./",count_file="Adult_Mouse_Brain_Coronal_Section_filtered_feature_bc_matrix.h5") #导入空间转录组数据
#由于靶场文件名64位的限制，h5文件做了重命名，如果在本地运行的导入下载的数据，代码为:
adata=sc.read_visium(path="./",count_file="V1_Adult_Mouse_Brain_Coronal_Section_2_filtered_feature_bc_matrix.h5")
adata.var_names_make_unique() #对重复的基因名称进行修改，确保每个基因名称都是唯一的
adata #查看数据内容
# 检查数据
adata.var["mt"] = adata.var_names.str.match("^MT-|^mt-") #标记线粒体基因，线粒体基因的命名通常以"MT-"或"mt-"开头
sc.pp.calculate_qc_metrics(adata,qc_vars=["mt"],inplace=True) #计算线粒体质控指标
adata.var["RP"]=adata.var_names.str.match("^RPS|^RPL|^Rps|^Rp1") #标记核糖体基因
sc.pp.calculate_qc_metrics(adata,qc_vars=["RP"],inplace=True) #计算核糖体质控指标
fig, axs = plt.subplots(1, 4, figsize=(16, 4)) #创建一个包含1行4列的图形，并设置整个图形的尺寸为16x4英寸
sns.distplot(adata.obs["total_counts"], kde=False,bins=60, ax=axs[0]) #绘制total_counts(每个细胞中的总计数)分布图，kde=False表示不显示核密度估计曲线,bins=60表示将数据分成
60个区间
sns.distplot(adata.obs["n_genes_by_counts"], kde=False, bins=60, ax=axs[1]) #n_genes_by_counts(每个细胞中检测到的基因数量)分布图
sns.distplot(adata.obs["pct_counts_mt"], kde=False,bins=60, ax=axs[2]) #与线粒体相关的基因在总基因计数中所占的比例
sns.distplot(adata.obs["pct_counts_RP"], kde=False, bins=60, ax=axs[3]) #与核糖体蛋白相关的基因在总基因计数中所占的比例。
plt.savefig('QC1.png')
```

代码示例

```
#数据质控,细胞过滤
sc.pp.filter_cells(adata,min_counts=5000) #过滤掉总计数小于5000的细胞
sc.pp.filter_cells(adata,max_counts=30000) #过滤掉总计数大于30000的细胞
sc.pp.filter_cells(adata, min_genes=2000) #保留具有至少2000个基因的细胞
sc.pp.filter_cells(adata, max_genes=7000) #保留具有最多7000个基因的细胞
adata=adata[adata.obs["pct_counts_mt"]<20] #保留线粒体相关基因比例小于20%的细胞
adata=adata[adata.obs["pct_counts_RP"]<5] #保留核糖体蛋白相关基因比例小于5%的细胞
fig, axs = plt.subplots(1, 4, figsize=(16, 4)) #过滤后的作图
sns.distplot(adata.obs["total_counts"], kde=False,bins=60, ax=axs[0])
sns.distplot(adata.obs["n_genes_by_counts"], kde=False, bins=60, ax=axs[1])
sns.distplot(adata.obs["pct_counts_mt"], kde=False,bins=60, ax=axs[2])
sns.distplot(adata.obs["pct_counts_RP"], kde=False, bins=60, ax=axs[3])
plt.savefig('QC2.png')
#基因过滤
sc.pp.filter_genes(adata, min_counts=1) #过滤掉在所有细胞中表达计数小于1的基因
sc.pp.filter_genes(adata, min_cells=10) #过滤掉在少于10个细胞中检测到的基因
sc.pl.highest_expr_genes(adata, n_top=20,save='highest_expr_genes.png') #显示具有最高表达水平的20个基因
#数据预处理
sc.pp.normalize_total(adata, inplace=True) # 归一化
sc.pp.log1p(adata) # 取对数
sc.pp.highly_variable_genes(adata, flavor="seurat", n_top_genes=2000) # 选择具有最高变异性的前2000个基因
sc.pl.highly_variable_genes(adata, save='highly_variable_genes.png') # 可视化高变异基因
adata=adata[:,adata.var.highly_variable] #保留具有高变异基因表达的细胞
```

代码示例

```
# 降维
sc.pp.pca(adata) #主成分分析(PCA),将高维的基因表达数据转换为低维的主成分
sc.pl.pca_variance_ratio(adata, save='pca_variance_ratio.png') #可视化PCA的方差比例
sc.pp.neighbors(adata)#计算细胞之间的相似性，构建近邻关系
sc.tl.tsne(adata) #是使用t-SNE算法对降维后的数据进行可视化
sc.tl.umap(adata) #使用UMAP算法对降维后的数据进行可视化

#聚类、结果可视化、marker基因
sc.tl.leiden(adata, key_added="clusters") #使用Leiden算法对单细胞RNA测序数据进行聚类分析
sc.pl.tsne(adata, color="clusters",save='tsne_clusters.png') #使用UMAP算法对聚类结果进行可视化
sc.pl.umap(adata, color="clusters",save='umap_clusters.png') #使用UMAP算法对聚类结果进行可视化
sc.pl.umap(adata, color='clusters',size=10,legend_loc='on data') #聚类标签在图上显示
sc.pl.spatial(adata, img_key="hires", color="clusters",save='spatial_clusters.png') #聚类的细胞进行空间可视化
sc.pl.spatial(adata, img_key="hires", color="clusters", groups=["0", "3"], crop_coord=[10000,20638,0, 10000], alpha=0.9, size=1.5,save='spatial_clusters_part.png') #特定的区域可视化
sc.tl.rank_genes_groups(adata, "clusters", method="t-test") #检测marker基因
sc.pl.rank_genes_groups_heatmap(adata, groups="5", n_genes=10, groupby="clusters",save='rank_genes_heatmap.png') #绘制差异基因的热图
sc.pl.rank_genes_groups_dotplot(adata, groups="5", n_genes=10, groupby="clusters",save='rank_genes_dotplot.png') #绘制差异基因的点图
```

代码示例

```
# 空间高可变基因
import SpatialDE #导入SpatialDE库,空间转录组数据的差异表达分析
import pandas as pd #导入pandas库，处理和分析数据。
import numpy as np # 导入numpy库，用于进行数值计算和数组操作
matrix=np.mat(adata.X.todense()) #将Anndata对象中的基因表达矩阵转换为numpy矩阵
counts = pd.DataFrame(matrix, columns=adata.var_names, index=adata.obs_names) #将基因表达矩阵转换为pandas DataFrame，其中行索引为adata.obs_names，列索引为adata.var_names。
coord = pd.DataFrame(adata.obsm['spatial'], columns=['x_coord', 'y_coord'], index=adata.obs_names) # 空间坐标数据转换为pandas DataFrame，其中行索引为adata.obs_names，列名为'x_coord'和'y_coord'。
results = SpatialDE.run(coord, counts) # 调用SpatialDE库中的run函数，对空间转录组数据进行差异表达分析，返回结果。
results.index = results["g"]#结果DataFrame中的'g'列设置为索引。
del results["g"]#删除结果DataFrame中的'g'列
results=results[['pval', 'qval','l', 'max_delta', 'max_ll', 'max_mu_hat','max_s2_t_hat', 'model', 'n', 's2_FSV', 's2_logdelta', 'time', 'BIC','max_ll_null', 'LLR','FSV','M']] #重新排列结果DataFrame的列的顺序
results=results.sort_values('qval') # 按照'qval'列的值对结果DataFrame进行排序。
print(results) #查看result
sc.pl.spatial(adata, img_key="hires", color=["Kcnab3", "Nfia"], alpha=0.7) #绘制基因空间表达分布
results.to_csv('top100.spatial.markers.tsv',sep='\t') # 生成空间高可变基因列表文件
```

China National GeneBank 国家基因库

欢迎访问：db.cngb.org

谢 谢

THANKS