**实验三，单细胞聚类分析**

**1. 实验内容**

本实验的主要目标是对单细胞测序数据进行聚类分析，具体内容包括：

读取单细胞测序数据及其对应的真实标签；

对高维数据进行降维处理，保留关键特征；

使用 KMeans 算法进行聚类分析；

通过 NMI 和 AMI 指标评估聚类效果；

可视化聚类结果，直观展示不同聚类簇的分布情况。

1. **分析及设计**

单细胞测序数据是一个高维数据集，基因的特征数量远超细胞样本数量。因此直接进行聚类可能会因噪声过多和维度灾难问题导致效果较差。

降维是处理高维数据的重要步骤，PCA（主成分分析）可将原始数据投影到低维空间，同时保留尽可能多的信息。

（1）实验过程：

数据读取与预处理：

首先读取数据文件（单细胞基因表达矩阵），将数据转置为每行对应一个细胞样本，列为基因特征；读取对应的真实标签，用于后续聚类效果评估。

（2）降维：

使用 PCA 方法对高维数据降维，保留 95% 的累计方差，减少噪声和计算复杂度；输出降维后的数据矩阵，作为聚类算法的输入。

（3）聚类分析：

选用 KMeans 聚类算法，根据真实标签的类别数设置聚类簇数；对降维后的数据进行聚类，得到每个细胞样本的聚类结果。

（4）聚类效果评估：

计算 NMI（归一化互信息）和 AMI（调整后的互信息）指标，评估聚类结果与真实标签的一致性；指标越高，说明聚类效果越好。

（5）可视化：

将聚类结果在屏幕上可视化，用颜色区分不同聚类簇。

1. **详细实现**

关键部分核心代码：

数据预处理：

使用pandas读取数据

# 读取基因数据和标签

#以zeisel.rds为例：

data = pd.read\_csv('zeisel.rds', sep='\t') # 数据

label = pd.read\_csv('zeisel.rds\_label', sep='\t') # 标签

# 数据转置，行样本，列特征

data\_T = data.T

降维

利用 sklearn 的 PCA 模块，将高维基因表达数据降维：

# PCA降维，保留95%的累计方差

pca = PCA(0.95)

pca.fit(data\_T)

X = pca.transform(data\_T) # 降维后

print(f"Data transformed by PCA. Shape: {X.shape}") # 查看

聚类

选用 sklearn 的 KMeans 层次聚类和DBSCAN 聚类算法，根据实际数据的类别数设置聚类簇数：

# --------------------- 聚类方法 1，KMeans ---------------------

n\_clusters = 9 # 类别数

km = KMeans(n\_clusters=n\_clusters, random\_state=1)

km.fit(X)

y\_predict = km.predict(X)

# --------------------- 聚类方法 2：层次聚类 ---------------------

# 使用 AgglomerativeClustering

hierarchical = AgglomerativeClustering(n\_clusters=n\_clusters, linkage='ward')

y\_predict\_hierarchical = hierarchical.fit\_predict(X)

# --------------------- 聚类方法 3：DBSCAN ---------------------

# DBSCAN 聚类（需调整参数以适应不同数据集）

dbscan = DBSCAN(eps=2, min\_samples=10) # 可调整eps（邻域半径）和min\_samples（最小样本数）

y\_predict\_dbscan = dbscan.fit\_predict(X)

效果评估

使用 NMI 和 AMI 指标评估聚类效果：

# 读取标签

label\_1 = label.to\_numpy()

label\_true = label\_1[:, 0] # 提取标签列

# 计算NMI和AMI指标

nmi\_score = normalized\_mutual\_info\_score(label\_true, y\_predict, average\_method='arithmetic')

ami\_score = adjusted\_mutual\_info\_score(label\_true, y\_predict, average\_method='arithmetic')

print(f"NMI: {nmi\_score:.4f}, AMI: {ami\_score:.4f}")

结果可视化

绘制聚类结果的二维散点图：

plt.scatter(X[:, 0], X[:, 1], c=y\_predict, cmap='viridis', s=10) # 使用颜色区分簇

plt.title("KMeans Clustering Result")

plt.xlabel("Principal Component 1")

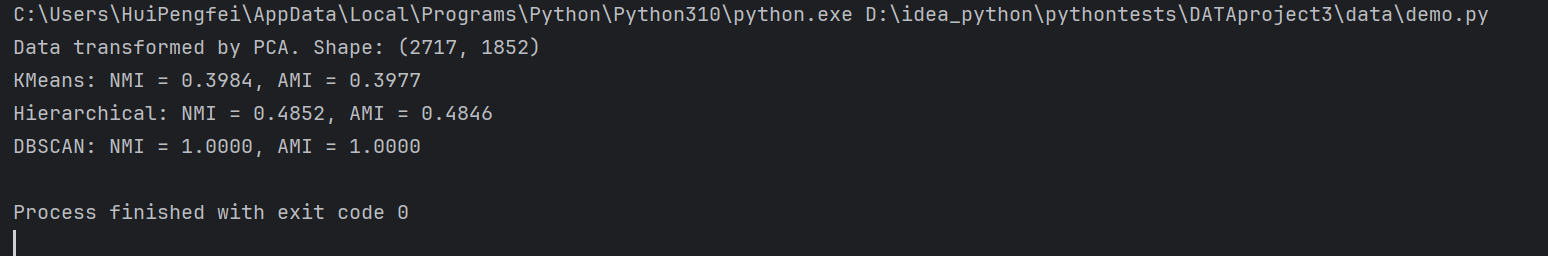
plt.ylabel("Principal Component 2")

plt.colorbar(label="Cluster ID")

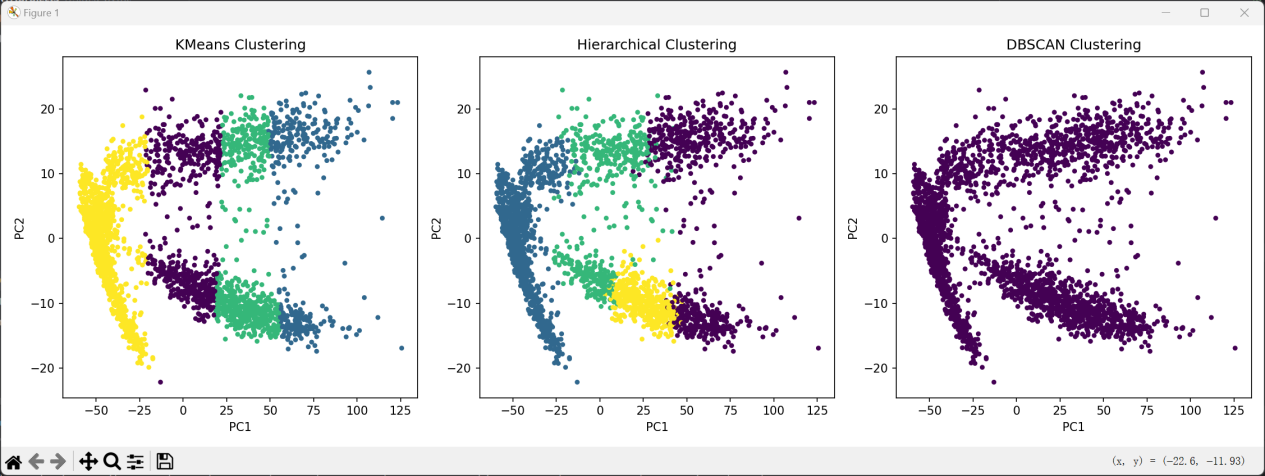
plt.show()

1. **实验结果**

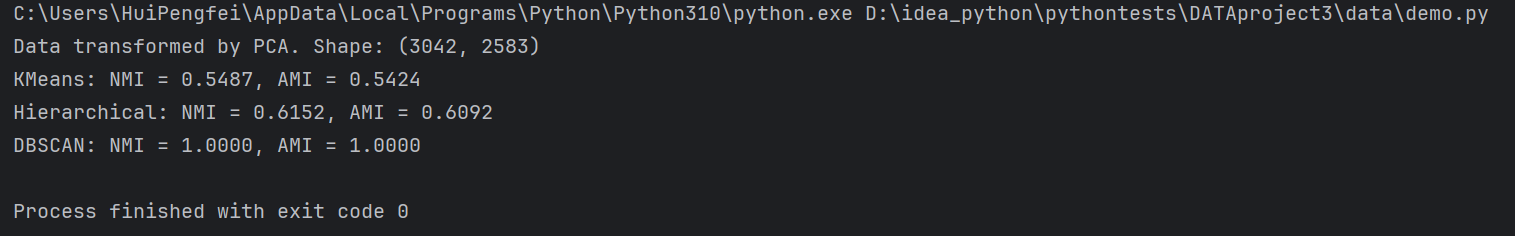
Klein：



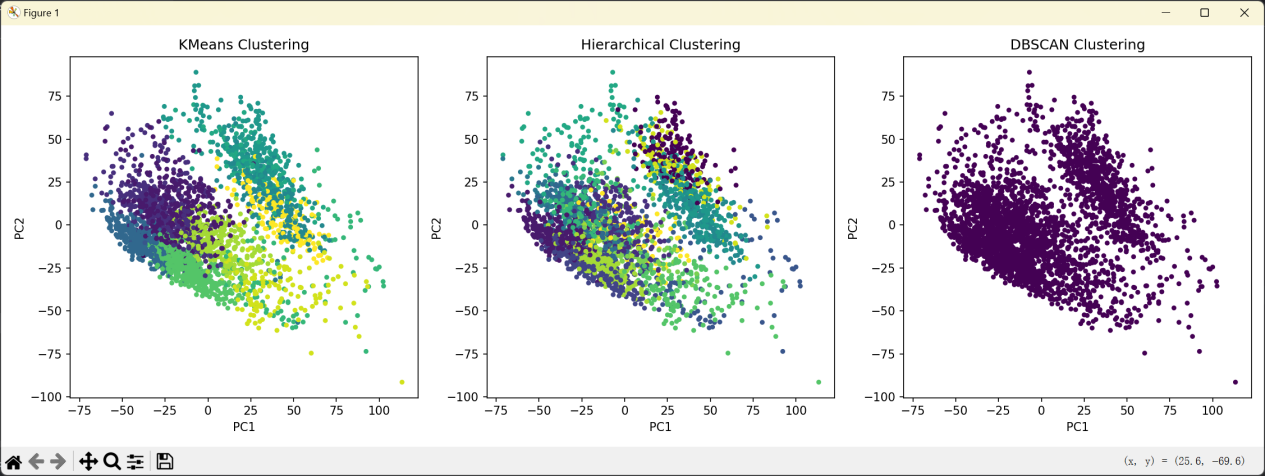
可视化：



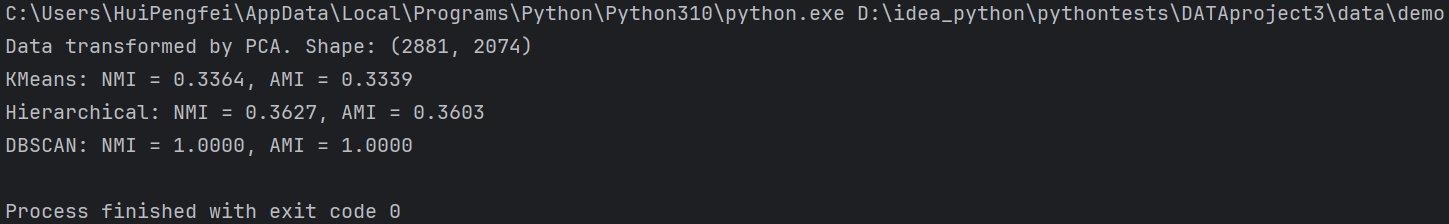
Lake：



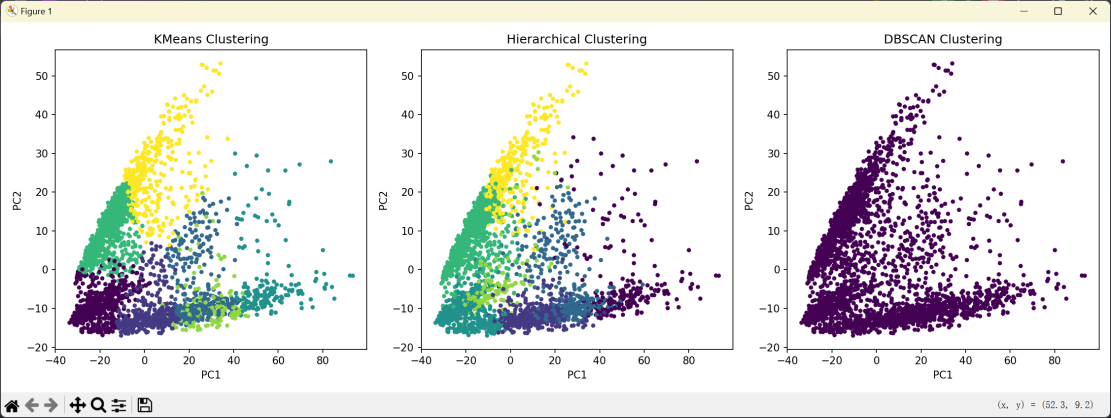
可视化：



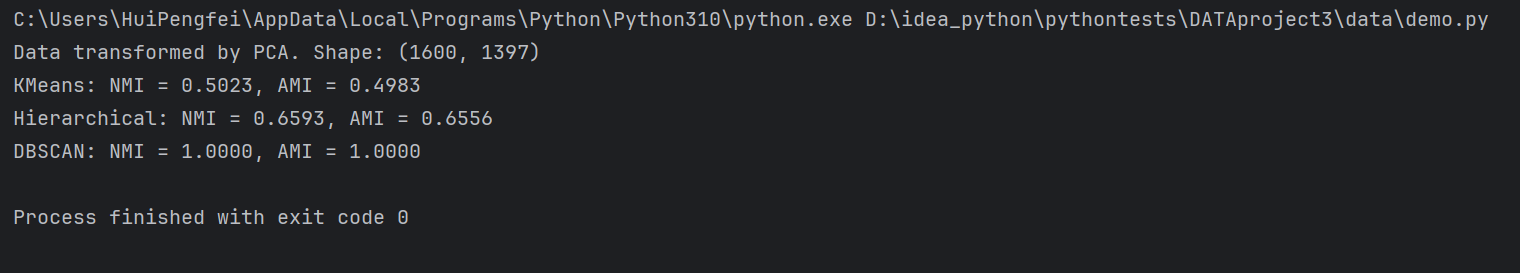
Romanov：



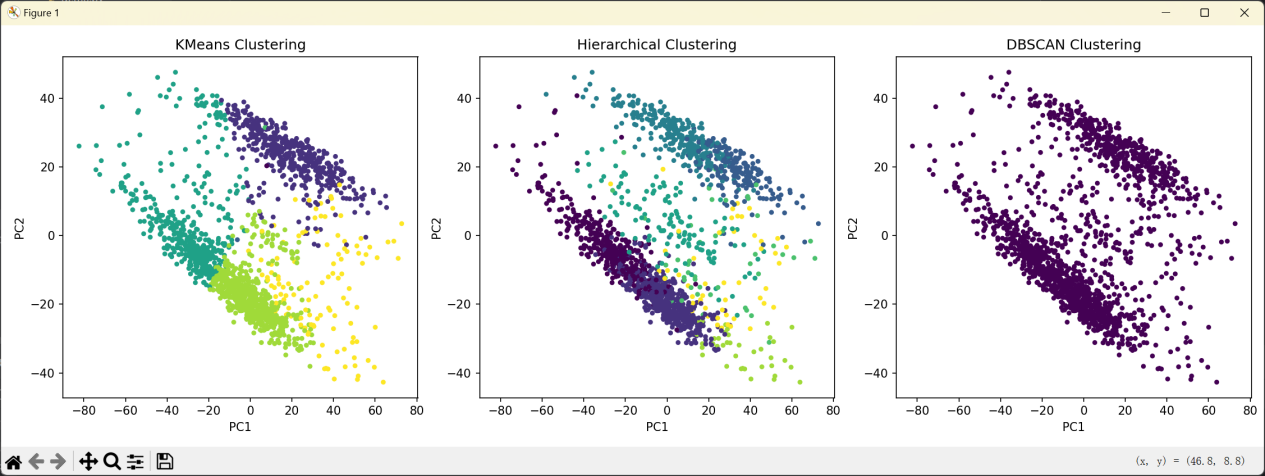
可视化：



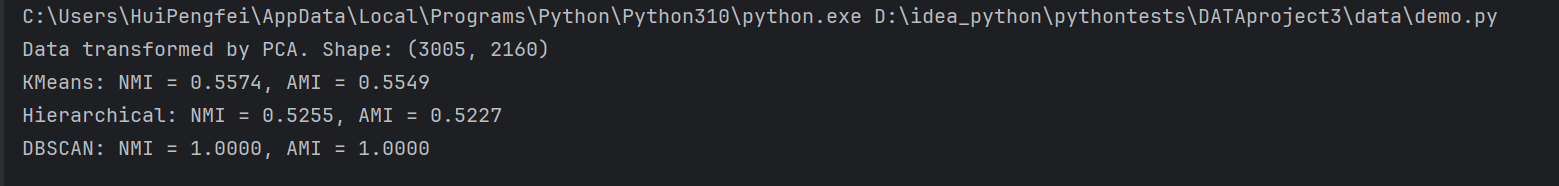
Xin：



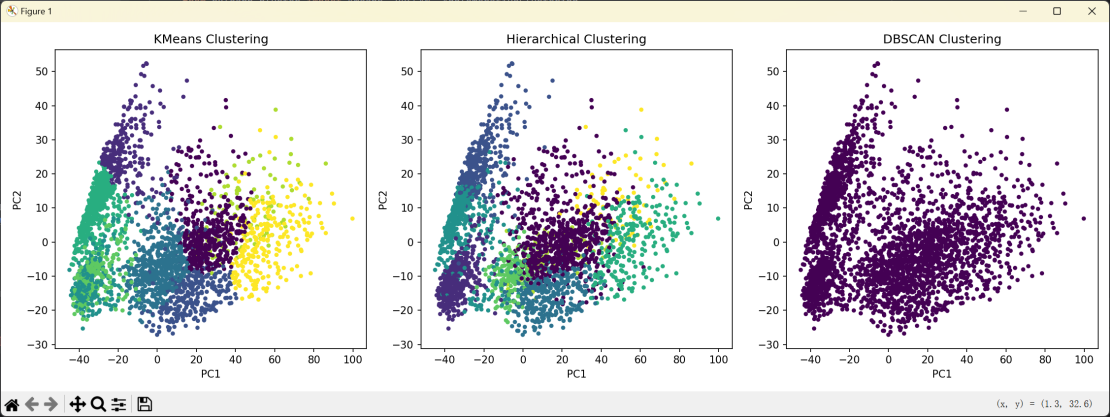
可视化：



Zeisel：



可视化：



1. **心得体会**

学习了 PCA 在降维中的作用，以及如何使用 Python 对高维数据进行降维处理。熟悉了 KMeans，层次聚类和DBSCAN聚类算法，的应用，并掌握了 NMI 和 AMI 两种指标的评估方法。理解了单细胞数据的特点及其聚类分析的流程。在数据处理时，刚开始使用R处理数据，但是总是因为无法有效的读取数据出BUG，在网上查阅资料以后发现直接使用pandas也可以读取数据，所以在遇到问题时尝试更多的解决方案可能会更有效地解决问题，避免大量浪费时间，