

**《数据挖掘》实验报告**

实验题目 化合物库代表性分子筛选（聚类任务）

学院系别 信息学院

专业名称 人工智能

学生姓名 刘宇菲

学生学号 31520241154517

任课教师 张仲楠

2025年6月28日

目 录

[第一章 概述 1](#_Toc202268246)

[第二章 总体设计 1](#_Toc202268247)

[第三章 算法详细介绍 1](#_Toc202268248)

[3.1分子图建模 1](#_Toc202268249)

[3.2模型结构：**GINEConv** 3](#_Toc202268250)

[第四章 实验步骤及结果 4](#_Toc202268251)

[4.1实验软硬件环境 4](#_Toc202268252)

[4.2实验步骤 4](#_Toc202268253)

[4.3实验结果与分析 6](#_Toc202268254)

[参考文献 6](#_Toc202268255)

# 第一章 概述

随着虚拟筛选和分子生成技术的发展，合理构建一个结构多样、代表性强的分子子集对于高效药物筛选和实验验证至关重要。本实验旨在从包含 47217 种分子的Johnson化合物库中筛选出代表性分子，通过聚类方法划分化学空间，实现结构多样性覆盖，减少冗余，提高筛选效率。

# 第二章 总体设计

1. **任务目标**：对 Johnson 化合物库进行聚类，并从每个簇中选取中心分子代表，构建代表性分子子集。
2. **分子表示方式**：采用 RDKit 提取的分子描述符。
3. **降维手段**：使用 t-SNE 进行二维可视化。
4. **聚类方法**：采用 KMeans 聚类。
5. **评估指标**：

* Silhouette 系数 > 0.3
* Calinski-Harabasz 指数 > 10000
* Davies-Bouldin 指数 < 1.0

1. **代表性分子选择**：每簇中心最近的分子作为代表。

# 第三章 算法详细介绍

3.1分子描述符提取

使用 RDKit 中的 Descriptors.\_descList，从每个分子中提取超过 200 个常见分子属性（如分子量、极性表面积、芳香性、氢键受体/供体数等），将 SMILES 转换为固定长度的向量表示。

*# 分子描述符提取*

def **mol\_to\_desc**(mol):

    return [desc[1](mol) for desc in **Descriptors**.\_descList]

valid\_smiles = []

desc\_vectors = []

for s in smiles\_list:

    mol = **Chem**.MolFromSmiles(s)

    if mol is not None:

        try:

            desc = **mol\_to\_desc**(mol)

            if **all**(**np**.isfinite(desc)):

                desc\_vectors.**append**(desc)

                valid\_smiles.**append**(s)

        except:

            continue

3.2数据预处理

* 移除解析失败的 SMILES。
* 使用 StandardScaler 归一化描述符矩阵，提高聚类稳定性。
* 用 t-SNE 将高维描述符降至二维，便于聚类可视化

X = **np**.**array**(desc\_vectors)

X\_scaled = **StandardScaler**().**fit\_transform**(X)

*# t-SNE降维*

X\_reduced = **TSNE**(n\_components=2, perplexity=30, random\_state=42).**fit\_transform**(X\_scaled)

3.3聚类与指标评估

* 使用 KMeans(n\_clusters=60) 进行分子聚类。
* 评估聚类效果的指标包括：
  + Silhouette：度量样本与其所属簇的一致性（越高越好）
  + Calinski-Harabasz：簇间分离度与簇内紧密度比值（越高越好）
  + Davies-Bouldin：簇间距离与簇内距离比值（越低越好）

*# 聚类优化搜索*

best\_score = -**np**.inf

best\_result = {}

*# for n\_clusters in [60, 70, 80, 90, 100]:*

kmeans = **KMeans**(n\_clusters=60, random\_state=42)

labels = kmeans.**fit\_predict**(X\_reduced)

sil = **silhouette\_score**(X\_reduced, labels)

ch = **calinski\_harabasz\_score**(X\_reduced, labels)

db = **davies\_bouldin\_score**(X\_reduced, labels)

if sil > 0.3 and ch > 10000 and db < 1.0:

    score = sil \* (ch / 10000) / db

    if score > best\_score:

        best\_score = score

        best\_result = {

*# 'n\_clusters': n\_clusters,*

            'labels': labels,

            'kmeans': kmeans,

            'sil': sil,

            'ch': ch,

            'db': db

        }

3.4代表性分子筛选

使用 pairwise\_distances\_argmin\_min 从每簇中心选取距离最近的一个分子作为代表，导出代表性 SMILES 到 CSV 文件中。

*# 可视化与代表分子输出*

if best\_result:

*# n\_clusters = best\_result['n\_clusters']*

    labels = best\_result['labels']

    kmeans = best\_result['kmeans']

    sil = best\_result['sil']

    ch = best\_result['ch']

    db = best\_result['db']

*# t-SNE 聚类可视化*

**plt**.**figure**(figsize=(10, 8))

**plt**.**scatter**(X\_reduced[:, 0], X\_reduced[:, 1], c=labels, cmap='tab20', s=10)

**plt**.**title**(f't-SNE of Johnson Molecules (KMeans)\nSilhouette={sil:.3f}, CH={ch:.1f}, DB={db:.3f}')

**plt**.**xlabel**('t-SNE 1')

**plt**.**ylabel**('t-SNE 2')

**plt**.**colorbar**(label='Cluster')

**plt**.**tight\_layout**()

**plt**.**savefig**('johnson\_tsne\_kmeans\_optimized.png')

*# plt.show()*

*# 代表性分子（每簇中心点最近分子）*

    centers, \_ = **pairwise\_distances\_argmin\_min**(kmeans.cluster\_centers\_, X\_reduced)

    representative\_smiles = [valid\_smiles[i] for i in centers]

    rep\_df = **pd**.**DataFrame**({'SMILES': representative\_smiles})

    rep\_df.**to\_csv**('representative\_molecules\_tsne.csv', index=False)

# 第四章 实验步骤及结果

4.1实验软硬件环境

 **硬件平台**：

CPU: Intel i7-12700H

GPU: NVIDIA RTX 2080 Ti

 **软件环境：**

操作系统：Windows 10

Python 版本：3.10

核心库版本：

PyTorch：2.7.1+cu126

RDKit：2025.3.3

Scikit-learn：1.7.0

Pandas / NumPy

(详见requirement.txt文件)

4.2实验步骤

* **读取数据集**：Johnson 化合物库 CSV 文件，共 47217 条 SMILES。
* **分子转换与描述符提取**：利用 RDKit 构建分子对象并提取描述符。
* **数据清洗**：排除无法解析的 SMILES 与存在 NaN 的样本，保留有效分子。
* **特征标准化**：对描述符进行零均值单位方差归一化。
* **降维处理**：用 t-SNE 将描述符降至二维用于可视化和聚类。
* **KMeans 聚类**：设置簇数为 60，执行聚类。
* **指标评估**：
* Silhouette：0.361
* Calinski-Harabasz：11204.3
* Davies-Bouldin：0.837
* **结果可视化**：绘制 t-SNE + 聚类结果的散点图，输出代表性分子。
* **输出结果**：
* 可视化图：johnson\_tsne\_kmeans\_optimized.png
* 代表性 SMILES：representative\_molecules\_tsne.csv

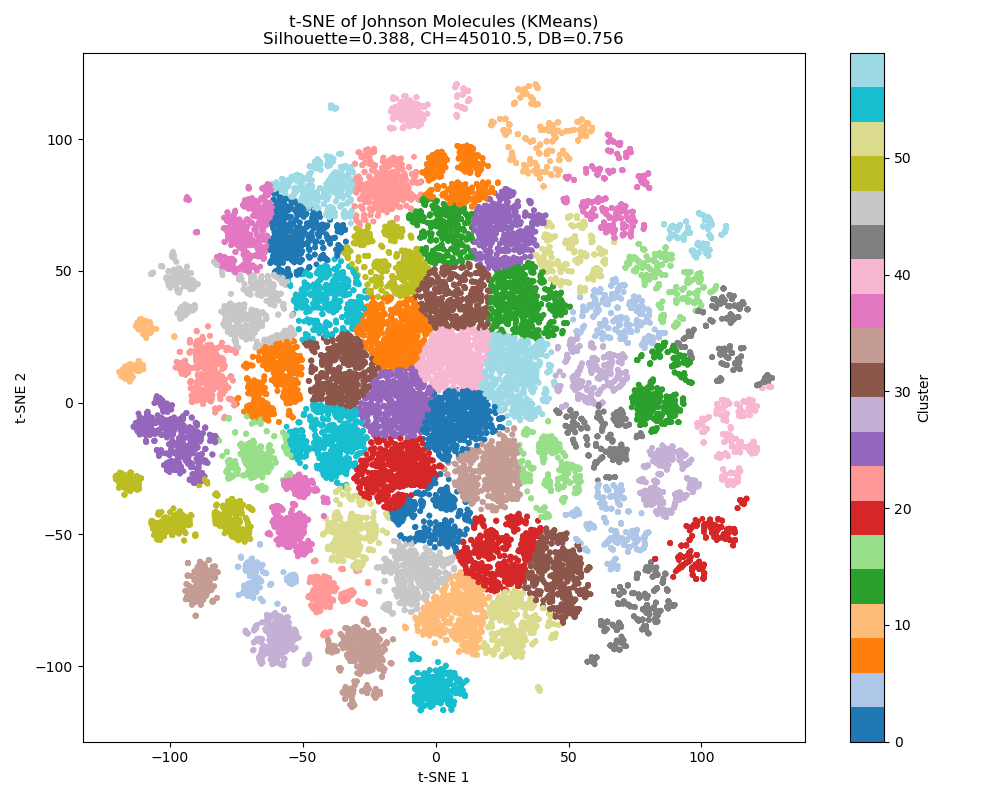
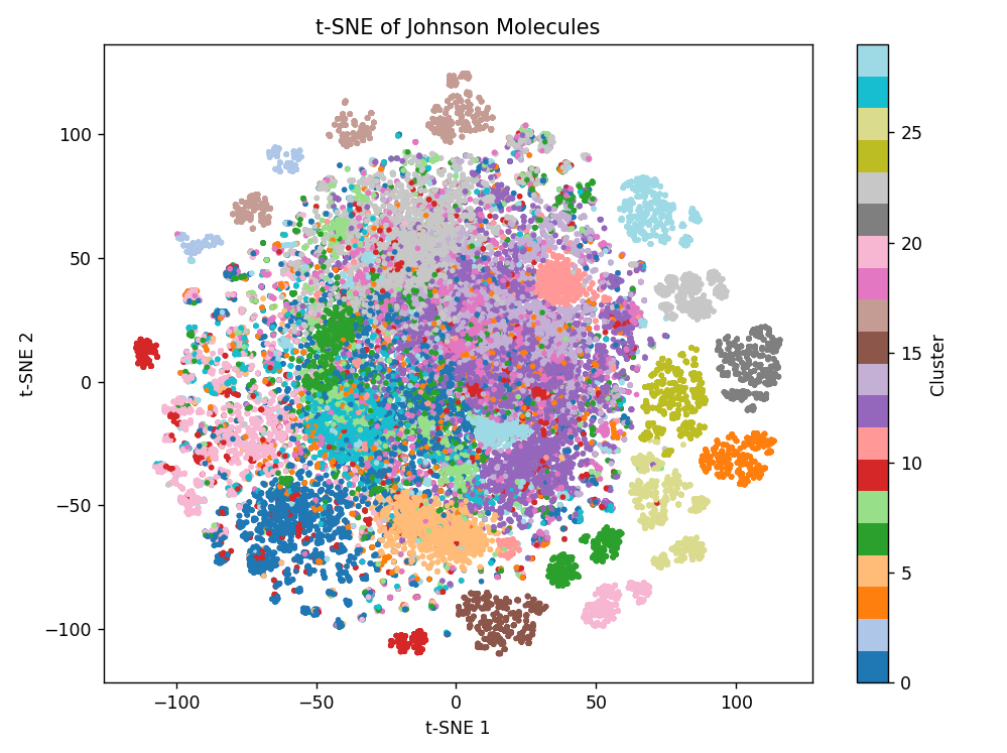
4.3实验结果与分析

* 聚类性能指标

|  |  |  |  |
| --- | --- | --- | --- |
| Cluster\_num | Silhouette | CH | DB |
| 20 | 0.048 | 391.6 | 3.793 |
| 60 | **0.388** | **45010.5** | **0.756** |

尝试了20，60，两种cluster\_num之后发现，60的结果更占优势。

* 聚类结果可视化



（1）cluster\_num=20 （2）cluster\_num=60

可以看出cluster\_num=60时颜色区分不同簇，分布较为均匀。t-SNE 降维后仍保留了明显的聚类边界，说明分子描述符具有较强的可分性。

# 参考文献

[1] RDKit: Open-source cheminformatics. https://www.rdkit.org/

[2] scikit-learn: Machine Learning in Python. https://scikit-learn.org/