中国研究生创新实践系列大赛

“华为杯”第十八届中国研究生

数学建模竞赛

学　　校	西南交通大学
参赛队号	21106130007
队员姓名	1.武姣熠
	2.徐潇
	3.刘健

中国研究生创新实践系列大赛

“华为杯”第十八届中国研究生

数学建模竞赛

题 目 抗乳腺癌候选药物的优化建模

摘 要:

乳腺癌是目前世界上最常见，致死率较高的癌症之一。本文的模型就挑选抗乳腺癌候选药物问题进行分析和建模。对于药物的筛选和实验具有一定的现实意义。本次建模主要提出了四个问题，分别是和化合物的ERα生物活性相关的分子描述符的筛选问题、化合物的ERα的生物活性预测问题、化合物的ADMET性质预测问题和分子描述符在什么取值或者处于什么取值范围时，能够使化合物对抑制ERα具有更好的生物活性，同时具有更好的ADMET性质的优化统计问题。

针对问题一，首先从附件给出的分子描述符分析入手，通过画图分析，发现中间存在一些干扰项，并且数据的量纲的影响问题。因此，先将分子描述符信息数据归一化，再将分子描述符的方差为零的分子描述符筛除，初步筛除了225个变量。由于分子描述信息较多，其与ERα生物活性的相关性就不能简单地认为是线性和非线性，综合了皮尔逊相关系数（线性）、斯皮尔曼秩相关系数（非线性）和距离相关系数（非线性）并进行了综合排序。由于变量之间还可能存在耦合，为此引入了PCA分析以及K-means聚类，根据PCA分析确定分组数，根据分组后组内相关性确定是否继续分组。最后建立了相关性综合评价模型和基于PCA与K-means算法的分子聚类模型。最后通过基于相关性指标与聚类结果的变量选择模型，较好地完成了20个主要变量的筛选。

针对问题二，在第一问筛选出的20个变量的基础上，建立了以adaBoost算法为基础的回归模型，首先对20个变量进行PCA分析，将训练集划出20%的数据作为模型验证的测试数据，拟合和优化参数之后，用所有的训练数据再次训练模型，最后对50个测试数据进行相同的数据处理后，用adaBoost预测模型完成了对化合物活性pIC50的预测，根据IC50和pIC50之间的负对数关系，计算得到IC50的值，较好地完成了化合物的活性预测，评估得到模型的平均绝对误差为0.55611。

针对问题三，主要考虑小样本的二分类问题，SVM和KNN算法对小样本学习具有很好的表现，并且模型较为简单，最终选定SVM和KNN算法建立，在这两个算法的基础上建立了基于PCA与SVC的ADMET预测模型和基于PCA与KNN的ADMET预测模型，较好地完成了对CaCo-2、HOB、CYP3A4、hERG和MN五种性质的预测。在验证集上分别取得100%、100%、93.92%、88.35%和94.94%的预测准确率。

针对问题四，首先对分子描述符多的问题，经过问题一的分析和验证，筛选出了20个和ERα生物活性相关性最高且相互之间相关性较低的分子描述符。因此，对ERα生物活性影响最大的分子描述符均在筛选出的20个变量中，其它相关性较小的分子描述符忽略，仅仅对筛选出的20个分子描述符进行分析。其次，对于问题约束较多的问题，对训练数

据集进行统计分析，最后建立基于统计分析和假设检验的模型，并统计出分子描述符对 ERα 生物活性和 ADMET 性质定量分析的模型。该模型简单，克服多目标优化算法复杂度大的问题。

关键词： PCA，相关分析，kmeans 聚类，adaBoost，SVM，KNN

1 问题重述

1.1 问题背景

乳腺癌是目前世界上最常见，致死率较高的癌症之一。乳腺癌的发展与雌激素受体密切相关，有研究发现，雌激素受体α亚型（Estrogen receptors alpha, ERα）在不超过10%的正常乳腺上皮细胞中表达，但大约在50%-80%的乳腺肿瘤细胞中表达；而对ERα基因缺失小鼠的实验结果表明，ERα确实在乳腺发育过程中扮演了十分重要的角色。目前，抗激素治疗常用于ERα表达的乳腺癌患者，其通过调节雌激素受体活性来控制体内雌激素水平。因此，ERα被认为是治疗乳腺癌的重要靶标，能够拮抗ERα活性的化合物可能是治疗乳腺癌的候选药物。比如，临床治疗乳腺癌的经典药物他莫昔芬和雷诺昔芬就是ERα拮抗剂。

目前，在药物研发中，为了节约时间和成本，通常采用建立化合物活性预测模型的方法来筛选潜在活性化合物，此外还需要在人体内具备良好的药代动力学性质和安全性，合称为 ADMET 性质。

1.2 需要解决的问题

问题1：针对1974个化合物的729个分子描述符进行变量选择，根据变量对生物活性影响的重要性进行排序，并给出前20个对生物活性最具有显著影响的分子描述符（即变量），并请详细说明分子描述符筛选过程及其合理性。

问题2：请结合问题1，选择不超过20个分子描述符变量，构建化合物对ERα生物活性的定量预测模型，请叙述建模过程。然后使用模型进行预测。

问题3：请利用提供的729个分子描述符，针对1974个化合物的ADMET数据，分别构建化合物的Caco-2、CYP3A4、hERG、HOB、MN的分类预测模型，并简要叙述建模过程。然后使用所构建的5个分类预测模型，对50个化合物进行相应的预测。

问题4：寻找并阐述化合物的哪些分子描述符，以及这些分子描述符在什么取值或者处于什么取值范围时，能够使化合物对抑制ERα具有更好的生物活性，同时具有更好的ADMET性质。

2 合理假设与符号系统

2.1 模型假设

（1）除题目提供的729个分子描述符外，没有其它影响生物活性的分子描述符；

（2）实验测定的1974个化合物对ERα的生物活性值IC_{50}、pIC_{50}准确无误或测量误差在可接受范围内；

（3）1974个化合物的ADMET性质对应值固定不变；

2.2 符号说明

序号	符号	解释
1	N	化合物个数
2	M	分子描述符个数

3	x_{ij}	第i个化合物第j个分子描述符值
4	$x_{ij}^{'}$	归一化后的第i个化合物第j个分子描述符值
5	$x_{ij}^{''}$	归一化后剔除无效因子的第i个化合物第j个分子描述符值
6	$x_{ij}^{'''}$	对生物活性最具有显著影响的第j个分子描述符第i个化合物值
7	X_j	第j个分子描述符列向量
8	$X_j^{'}$	归一化后的第j个分子描述符列向量
9	$X_j^{''}$	归一化后剔除无效因子的第j个分子描述符列向量
10	$X_j^{'''}$	第j个对生物活性最具有显著影响的分子描述符
11	$\bar{x}_{\cdot j}$	第j个分子描述符列向量的均值
12	y_{ij}	第i个化合物对 ERα 的生物活性值（j=1 对应 IC_{50}，j=2 对应 pIC_{50}）
13	Y_j	化合物对 ERα 的生物活性值 IC_{50}（j=1 对应 IC_{50}，j=2 对应 pIC_{50}）
14	$\bar{y}_{\cdot j}$	化合物对 ERα 的生物活性值的均值（j=1 对应 IC_{50}，j=2 对应 pIC_{50}）
15	Z_i	给定的五个 ADMET 性质中的第i个性质
15	$Z_i^{'}$	一致性转换后的五个 ADMET 性质中的第i个性质
16	r_1	皮尔逊线性相关系数
17	r_2	斯皮尔曼秩相关系数
18	r_3	距离相关系数
19	$Index_r$	综合相关指标
20	$rg(\cdot)$	秩次序函数
21	P_i	第i类分子描述符集合
22	$P_i^{'}$	第i个对生物活性最具有显著影响的分子描述符所在类
23	R_i^{min}	第i类分子描述符集合中各元素相关系数最小值
24	R_{ij}^{max}	第i类分子描述符集合与第j分子描述符集合间相关系数最大值
25	K_i^{pca}	第i类分子描述符集合主成分分析时，贡献率≥85%时的主成分个数
26	v_{i_s,i_l}	第i类分子描述符集合的协方差矩阵的特征向量

27	λ_{i_s,i_l}	第i类分子描述符集合的协方差矩阵的特征值
28	H	所有回归预测模型的集合
29	H_1	所有二分类模型的集合
30	$m_i^{'''}$	$X_j^{'''}$的最小值
31	$M_i^{'''}$	$X_j^{'''}$的最小值
32	$m_i^{''}$	$X_j^{''}$的最小值
33	$M_i^{''}$	$X_j^{''}$的最小值

3 问题一的建模与求解

3.1 问题分析

相关材料证明，乳腺癌与雌激素受体ERα密切相关，抗激素治疗常用于ERα表达的乳腺癌患者，通过调节雌激素受体活性来控制体内雌激素水平，在药物研发过程中，首先针对Erα收集一系列化合物和生物活性指标，以这一系化合物的分子结构作为自变量，将化合物的生物活性作为因变量，构建化合物定量结构-活性关系模型，然后根据该模型预测性能更好的化合物，实现药物优化。文件“Molecular_Descriptor. Xlsx”为各个化合物与化合物内部各分子含量数据，提供了1974个化合物和729个分子变量；文件“ERα_activity.xlsx”提供了各个化合物的生物活性。

根据生物活性筛选相关性较大的分子变量，可以将这一问题简化成相关性问题，有以下三个难点：

（1） 明显无相关性变量的干扰；

（2） 相关性分析模型的选取，无法直观判断分子自变量与化合生物活性因变量之间以及分子自变量之间是否为线性关系；

（3） 分子变量的选取，分子因变量之间存在信息重复问题；

针对这三个难点，做如下处理：

（1） 将原始数据做归一化处理后进行作图观察，观察各个变量的特点，剔除明显干扰项；

（2） 根据数据特点，选取皮尔森相关系数、斯皮尔曼相关系数、距离相关系数三个模型做数据分析，综合三个结果进行排序，得出综合排序表；

（3） 分子变量依照相关性分组，相关性强的为一组，每组选出一个作为分子变量代表，代表之间的相关性弱，减少信息重复；

3.2 模型建立

3.2.1 模型一：相关性综合评价模型

本题中，各分子描述符与化合物对ERα的生物活性值的关系未知，故选定皮尔森线性相关系数、斯皮尔曼秩相关系数、距离相关系数作为评估指标，建立相关性综合评价模型，评估各个分子描述符对生物活性影响的重要性。相关性综合评价模型如下[1-2]：

$$Index_r(X_i,Y_j)=\max_{1\le k\le 3}\left|r_k(X_i,Y_j)\right| \quad (1)$$

其中：

$$r_1(X_i,Y_j)=\frac{\sum_{k=1}^{N}(x_{ki}-\bar{x}_{\cdot i})(y_{kj}-\bar{y}_{\cdot j})}{\sqrt{\sum_{k=1}^{N}(x_{ki}-\bar{x}_{\cdot i})^2}\sqrt{\sum_{i=1}^{n}(y_{ij}-\bar{y}_{\cdot j})^2}} \tag{2}$$

$$r_2(X_i,Y_j)=1-\frac{6\sum d_s^2}{n(n^2-1)} \tag{3}$$

$$d_s=rg(x_{si})-rg(Y_{sj}) \tag{4}$$

$$r_1^2(X_i,Y_j)=\frac{v^2(X_i,Y_j)}{\sqrt{v^2(X_i,X_j)v^2(Y_j,Y_j)}} \tag{5}$$

$$v^2(X_i,Y_j)=\frac{1}{N^2}\sum_{l,s=1}^{N}A_{l,s}B_{l,s} \tag{6}$$

$$v^2(X_i,X_i)=\frac{1}{n}\sum_{l,s=1}^{n}A_{l,s}^2 \tag{7}$$

$$v^2(Y_j,Y_j)=\frac{1}{n}\sum_{l,s=1}^{n}B_{l,s}^2 \tag{8}$$

$$A_{l,s}=\|x_{li}-x_{si}\|_2-\frac{1}{n}\sum_{k=1}^{n}\|x_{ki}-x_{si}\|_2-\frac{1}{n}\sum_{g=1}^{n}\|x_{li}-x_{gi}\|_2+\frac{1}{n^2}\sum_{k,g=1}^{n}\|x_{ki}-x_{gi}\|_2 \tag{9}$$

$$B_{l,s}=\|y_{lj}-y_{sj}\|_2-\frac{1}{n}\sum_{k=1}^{n}\|y_{kj}-y_{sj}\|_2-\frac{1}{n}\sum_{g=1}^{n}\|y_{lj}-y_{gj}\|_2+\frac{1}{n^2}\sum_{k,g=1}^{n}\|y_{kj}-y_{gj}\|_2 \tag{10}$$

在统计学中，相关系数反映的相关性强弱如表1约定[3]，其绝对值约接近1，相关性越强，即该分子描述符对生物活性值的影响越重要，因此可根据各个分子描述符与生物活性值之间的综合相关性系数$Index_r$，将各个分子描述符对生物活性影响的重要性进行排序。

表1 相关程度度量表

相关性	相关系数的绝对值
极强相关	0.8~1.0
强相关	0.6~0.8
中相关	0.4~0.6
弱相关	0.2~0.4
极弱或无相关	0~0.2

3.2.2 模型二：基于PCA与K-means算法的分子聚类模型

问题一要求从多个分子变量中选前20个对生物活性最具有显著影响的分子描述符，然而题目中给出的部分分子描述符间本就存在极强的相关性，为了减少选取变量的信息重复程度，随即建立分子描述符聚类模型，根据相关程度将所有分子描述符分类，具体如下：

目标模型：

$$\{P_1,P_2,\cdots P_{M_0}\}=f_1(X_1,X_2,\cdots X_M) \tag{11}$$

约束条件：

（1）$P_i\cap P_j=\emptyset,\ \forall i\neq j$；

（2）$P_1\cup P_2\cup\cdots\cup P_{M_0}=\{X_1,X_2,\cdots X_M\}$；

（3）$R_i^{min} \geq 85\%$，$\forall i \in \{1,2,\cdots,M_0\}$；

（4）$R_{ij}^{max} < 85\%$，$\forall i \neq j$；

（5）$K_i^{pca} \leq 1$，$\forall i \in \{1,2,\cdots,M_0\}$。

其中，$\forall P_i = \left\{X_{i_1}, X_{i_2}, \cdots X_{i_{M_i}}\right\}, P_j = \left\{X_{j_1}, X_{j_2}, \cdots X_{j_{M_i}}\right\}$

$$R_i^{min} = \max\left\{\min_{1\leq s\leq l\leq M_i} r_1\left(X_{i_s}, X_{i_l}\right), \min_{1\leq s\leq l\leq M_i} r_2\left(X_{i_s}, X_{i_l}\right)\right\} \tag{12}$$

$$R_{ij}^{max} = \max\left\{\max_{1\leq s\leq M_i, 1\leq l\leq M_j} r_1\left(X_{i_s}, X_{j_l}\right), \max_{1\leq s\leq M_i, 1\leq l\leq M_j} r_2\left(X_{i_s}, X_{j_l}\right)\right\} \tag{13}$$

K_i^{pca}通过对P_i进行主成分分析得到[4]，具体操作如下：

第一步：计算平均值

$$\bar{x}_{\cdot i_k} = \frac{1}{N}\sum_{n=1}^{N} x_{n,i_k} \tag{14}$$

第二步：计算协方差矩阵$Cov_{P_i} = \left\{cov_{i_s,i_l}\right\}_{s,l=1}^{M_i}$

$$\begin{aligned} Cov\left(X_{i_s}, X_{i_l}\right) &= E\left[\left(X_{i_s} - E\left(X_{i_s}\right)\right)\left(X_{i_l} - E\left(X_{i_l}\right)\right)\right] \\ &= \frac{1}{N-1}\sum_{n=1}^{N}\left(x_{n,i_s} - \bar{x}_{\cdot i_s}\right)\left(x_{n,i_l} - \bar{x}_{\cdot i_l}\right) \end{aligned} \tag{15}$$

第三步：计算协方差矩阵的特征值与特征向量

$$Cov_{P_i} v_{i_s,i_l} = \lambda_{i_s,i_l} v_{i_s,i_l} \tag{16}$$

第四步：对特征值从大到小排序为$\lambda_{i_s,i_l}^{(1)}, \lambda_{i_s,i_l}^{(2)}, \cdots, \lambda_{i_s,i_l}^{(M_i)}$，计算贡献率，确定$K_i^{pca}$值：

$$\frac{\sum_{n=1}^{K_i^{pca}} \lambda_{i_s,i_l}^{(n)}}{\sum_{n=1}^{M_i} \lambda_{i_s,i_l}^{(n)}} \geq 0.85 \tag{17}$$

3.2.3 模型三：基于相关性指标与聚类结果的变量选择模型

模型一拟将评估各个分子描述符对生物活性影响的重要性，并基于此对分子描述符排序；模型二拟将所有分子描述符进行分类，形成类内变量高度相关、类外变量弱相关的多个分子描述符类别。根据题意，需筛选出前20个对生物活性最具有显著影响的分子描述符，此时，若仅根据模型一排序结果进行筛选，难免会选出具有极强相关性的多个变量，这会使得筛选分子描述符在某一方面信息冗余，但在另一方面信息缺失。因此本文结合模型一结果与模型二结果，建立了如下基于相关性指标与聚类结果的变量选择模型。

目标模型：

$$\left\{X_1''', X_2''', \cdots X_{20}'''\right\} = f_2\left(P_1, P_2, \cdots P_{M_0}, \left\{Index_r(X_i, Y_2)\right\}_{i=1,}^{N}\right) \tag{18}$$

约束条件：

（1）$Index_r\left(X_{20}''', Y_2\right) \leq Index_r\left(X_{19}''', Y_2\right) \leq \cdots \leq Index_r\left(X_1''', Y_2\right) = \max_{\forall X_i}\left\{Index_r(X_i, Y_2)\right\}$；

（2）$\forall X_i$，$X_i \in P_{i_0}$，若 $\max\limits_{X_k \in P_{i_0}} \{Index_r(X_k, Y_2)\} > Index_r\left(X_{20}^{'''}, Y_2\right)$，则 $P_{i_0} \in \left\{P_k^{'}\right\}_{k=1}^{20}$；

（3）$\forall i \in \{1,2,\cdots,20\}$，$X_i^{'''} \in P_i^{'}$，则 $\forall j \neq i$，$X_j^{'''} \notin P_i^{'}$；

（4）$\forall i \in \{1,2,\cdots,20\}$，若 $X_i^{'''} \in P_i^{'}$，则 $Index_r\left(X_i^{'''}, Y_2\right) = \max\limits_{X_k \in P_i^{'}} \{Index_r(X_k, Y_2)\}$

3.3 求解过程

3.3.1数据预处理

原始数据中分子变量数值分散，选取个别变量作图为例，如图1所示。

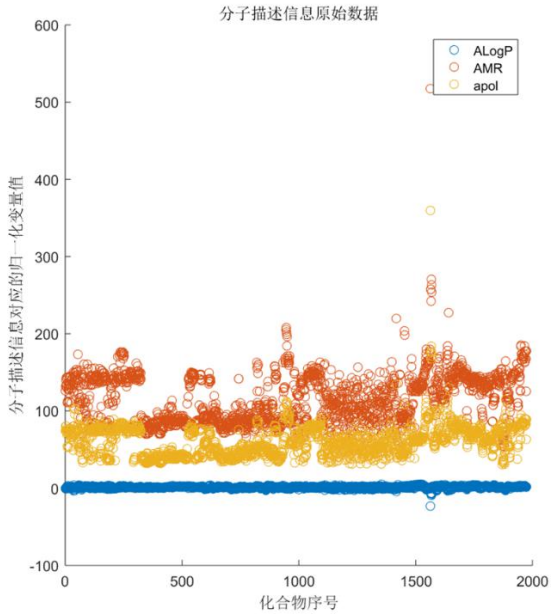

图1 归一化前分子变量值分布案例

图中三个分子变量ALogP、AMR、apol的值分别大致分布在区间(100，200)、区间(0，100)与区间(-10，10)之间，不便于数据分析。归一化处理保持数据分布情况形状不变的同时将数据集中起来。归一化处理公式如下：

$$y = \frac{(y_{max} - y_{\min}) * (x - x_{min})}{x_{max} - x_{\min}} + y_{min} \tag{19}$$

如图2所示。

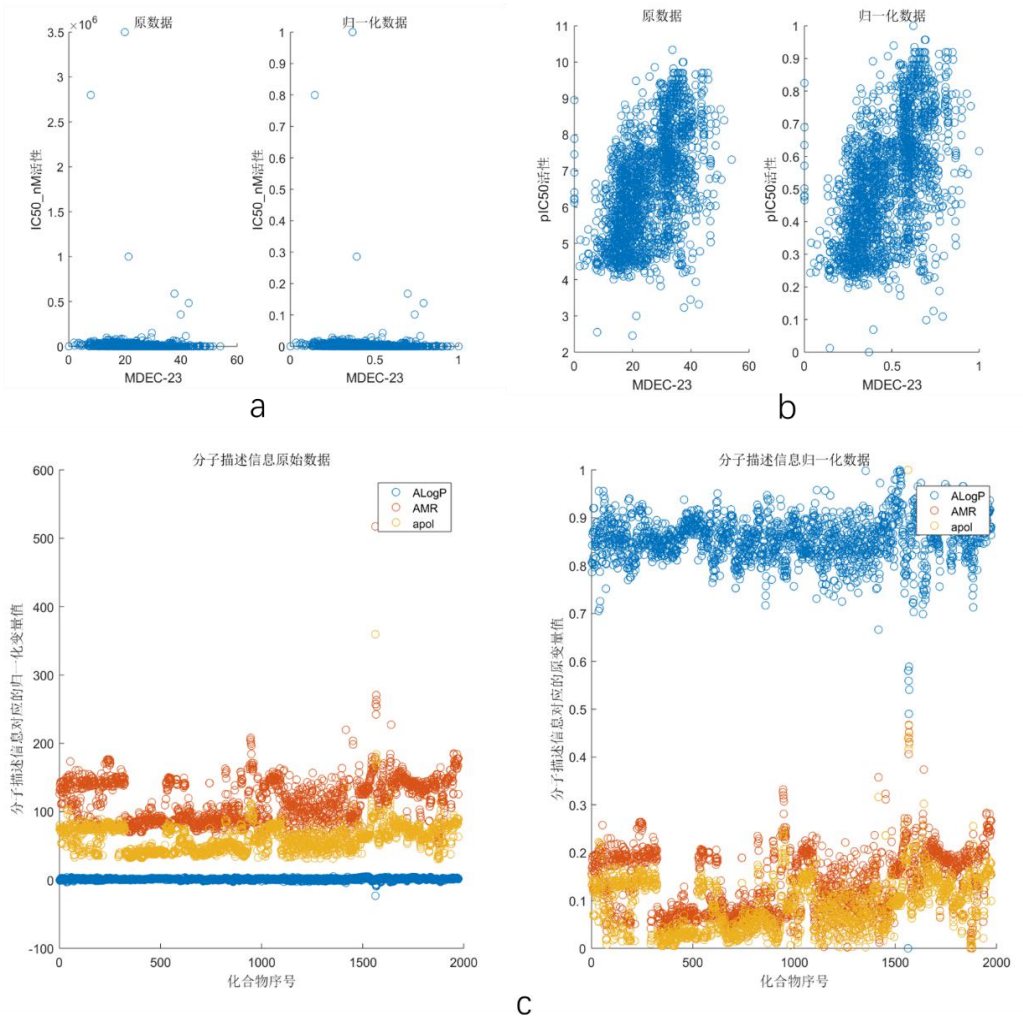

图2 归一化前后对比图

a、b两图显示对于同一个分子，在归一化后，分子变量的值集中在区间$(0，1)$，并且同一个分子变量前后值的分布形状没有改变；c图显示不同分子在归一化后消除量纲影响，均分布在区间$(0，1)$。

归一化处理后，化合物活性值的分布如图3所示。

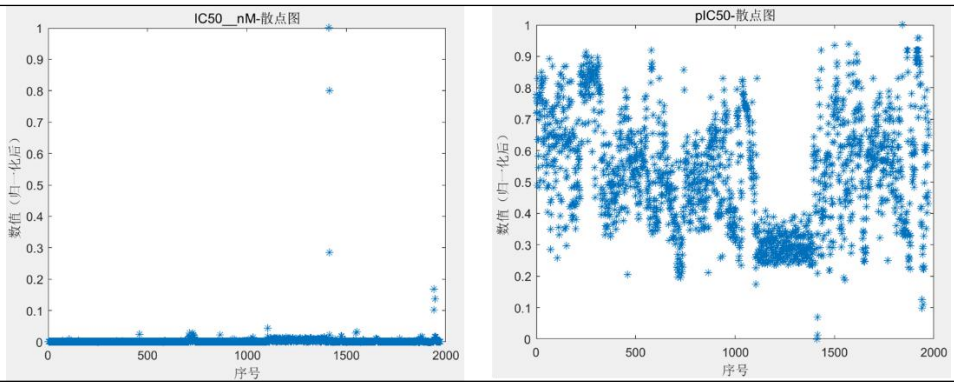

图3 1974个化合物活性分布图

化合物活性IC50_nM值在y轴上的投影过于集中，说明数据在某一维度上被压缩，相比之下，pIC50的值分布分散，有更明显的相关性。选取具体分子变量做对比分析，结果如

图4所示。

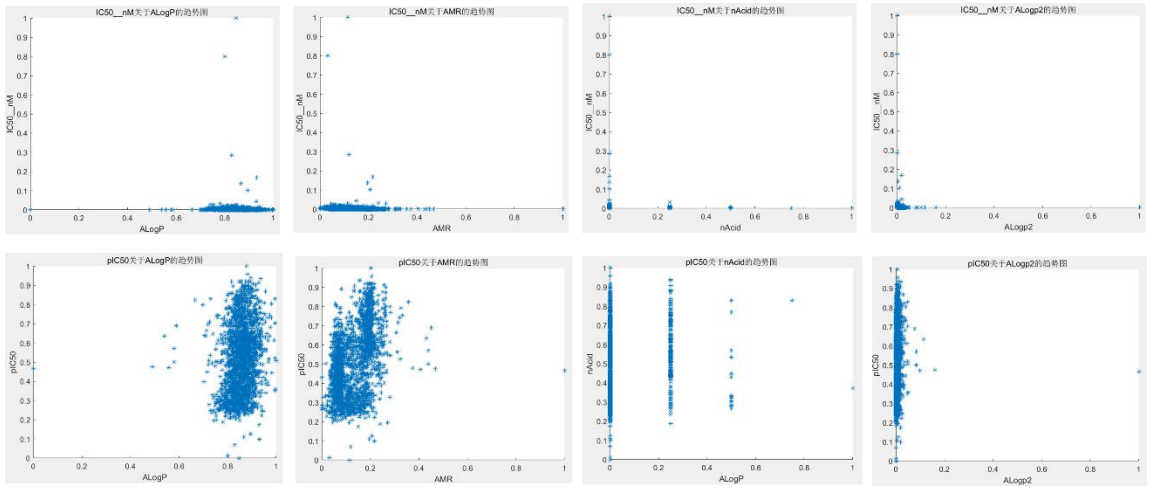

图4 活性IC50_nM、pIC50关于分子ALogP、AMR、nAcid、ALogp2的趋势图

图中第一行为活性IC50_nM关于分子ALogP、AMR、nAcid、ALogp2的趋势图，第二行为pIC50关于分子ALogP、AMR、nAcid、ALogp2的趋势图，pIC50明显具有更好的性质。

故而在化合物活性度量上选取pIC50，本章节分子变量作为自变量，pIC50作为唯一因变量。个别分子为明显干扰项，如图5所示。

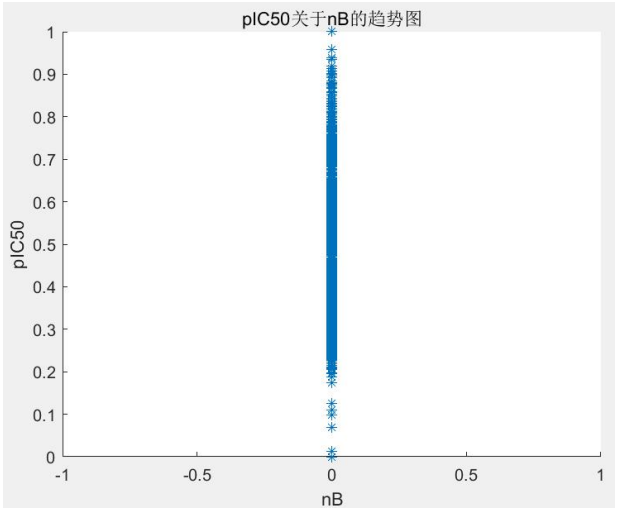

图5 原数据中的明显干扰项

图中所展示的分子nB的值一直为0，pIC50的值大小与之无关，为干扰项。

将每个分子变量求方差，方差用于衡量一组数据的波动大小，方差为0，则说明该组数据没有波动，证明与化合物的生物活性无关，方差公式为：

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n} \tag{20}$$

对分子变量求方差的结果如图6所示。

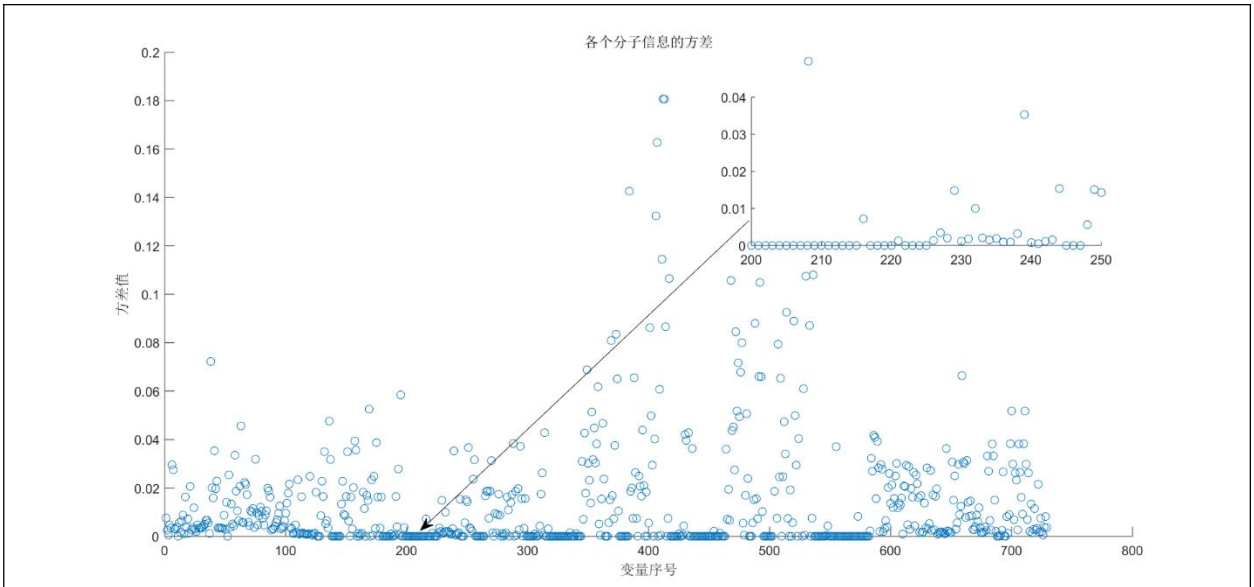

图6 分子变量求方差的结果

图中多个分子变量的方差为0，将该分子变量剔除，得到样本数据1。

3.3.2 模型一求解过程

3.3.2.1算法

分子与化合物活性相关性综合评价模型算法如图7所示，选取皮尔森线性相关系数、斯皮尔曼秩相关系数、距离相关系数三者绝对值的最大值进行排序。

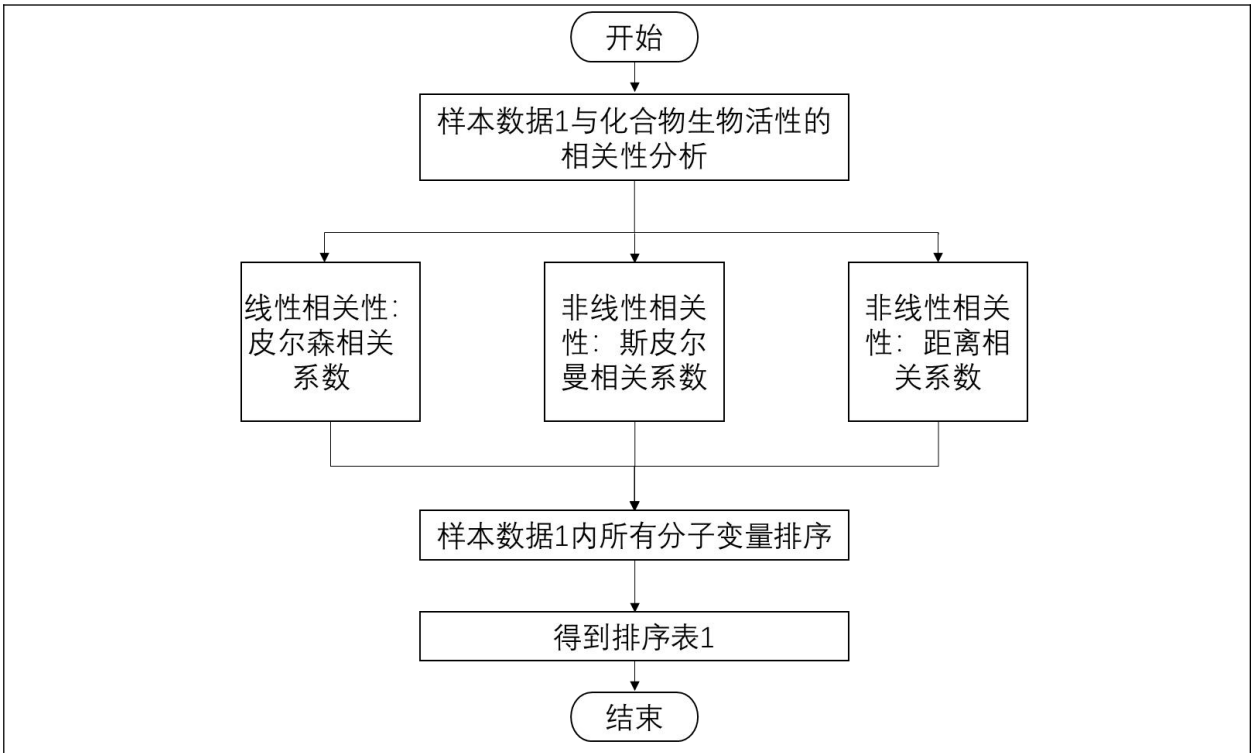

图7 相关性综合评价模型算法

3.3.2.2结果

采用相关性分析模型对样本数据1进行分析，综合三个分析结果，得到排序表，如表2所示，表中为排序表1的前15个分子。

表2 排序表

顺序	分子	皮尔森线性相关系数	斯皮尔曼秩相关系数	距离相关系数	三者绝对值最大值
1	MDEC-23	0.538047798	0.549051863	0.542485087	0.549051863
2	MLogP	0.529321142	0.544950865	0.533140314	0.544950865
3	LipoaffinityIndex	0.491854942	0.524904722	0.514648216	0.524904722
4	C1SP2	0.406928649	0.502389663	0.486270991	0.502389663
5	nC	0.459548956	0.486766211	0.49913269	0.49913269
6	maxsOH	0.466620545	0.461923285	0.480184606	0.480184606
7	minsOH	0.466127282	0.426563533	0.475877712	0.475877712
8	CrippenLogP	0.412300157	0.473745742	0.454678011	0.473745742
9	AMR	0.425149034	0.454511405	0.471977308	0.471977308
10	minHsOH	0.399129873	0.18819736	0.466701346	0.466701346
11	maxHsOH	0.408760763	0.233509234	0.465326851	0.465326851
12	apol	0.383253108	0.440487026	0.462082368	0.462082368
13	ATSp5	0.383628006	0.4514054	0.45871478	0.45871478
14	hmin	0.426365028	0.42664839	0.458561862	0.458561862
15	nBonds2	0.374058099	0.435046602	0.453506895	0.453506895

3.3.3模型二求解过程

3.3.3.1算法

问题一要求从多个分子变量中选取20个，为了减少选取变量的信息重复程度，对全部分子便变量按照相关性程度进行分类，分类前采用PCA分析决定分组类别，再使用K-means分类算法[5]，进行聚类，对分出的组别经行组内部相关性分析，分析结果有以下 种情况：

（1） 若组内分子自变量相关性小于阈值，且该组别与化合物生物活性相关性强，则对该组经行进一步细分；

（2） 若组内分子自变量相关性小于阈值，且该组别与化合物生物活性相关性弱，则丢弃该组别；

（3） 若组内分子自变量相关性大于阈值，且该组别与化合物生物活性相关性强，则保留该组别且无需再细分

（4） 若组内分子自变量相关性大于阈值，且该组别与化合物生物活性相关性弱，则丢弃该组别；

分子自变量分组的思路算法如图8所示。

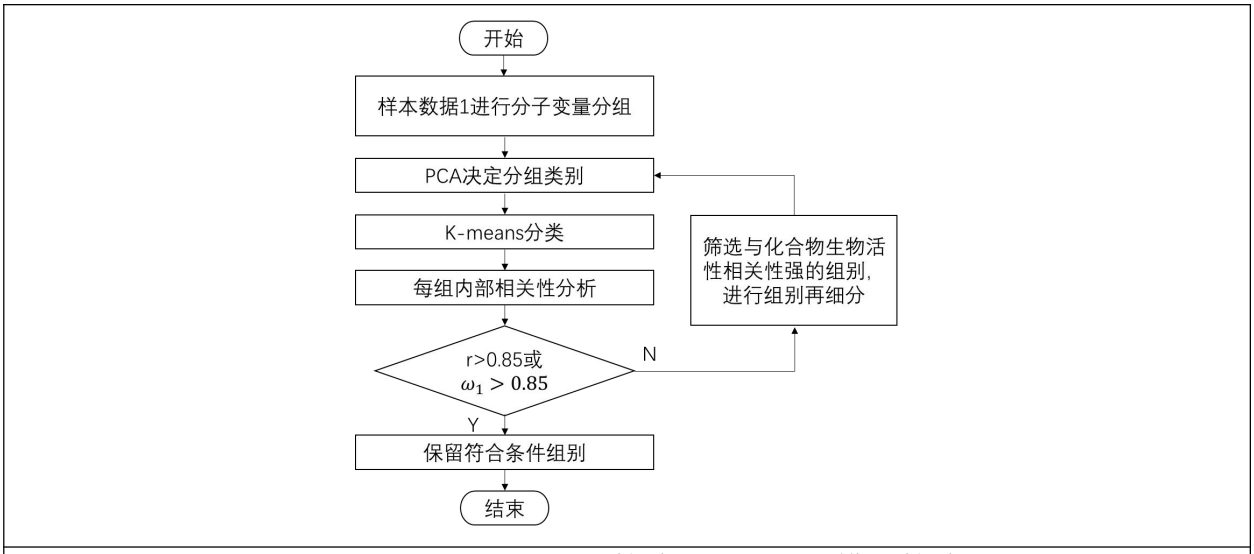

图8 基于PCA与K-means算法的分子聚类模型算法

3.3.3.2 求解

模型二进行组别分类，具体过程如图9所示。

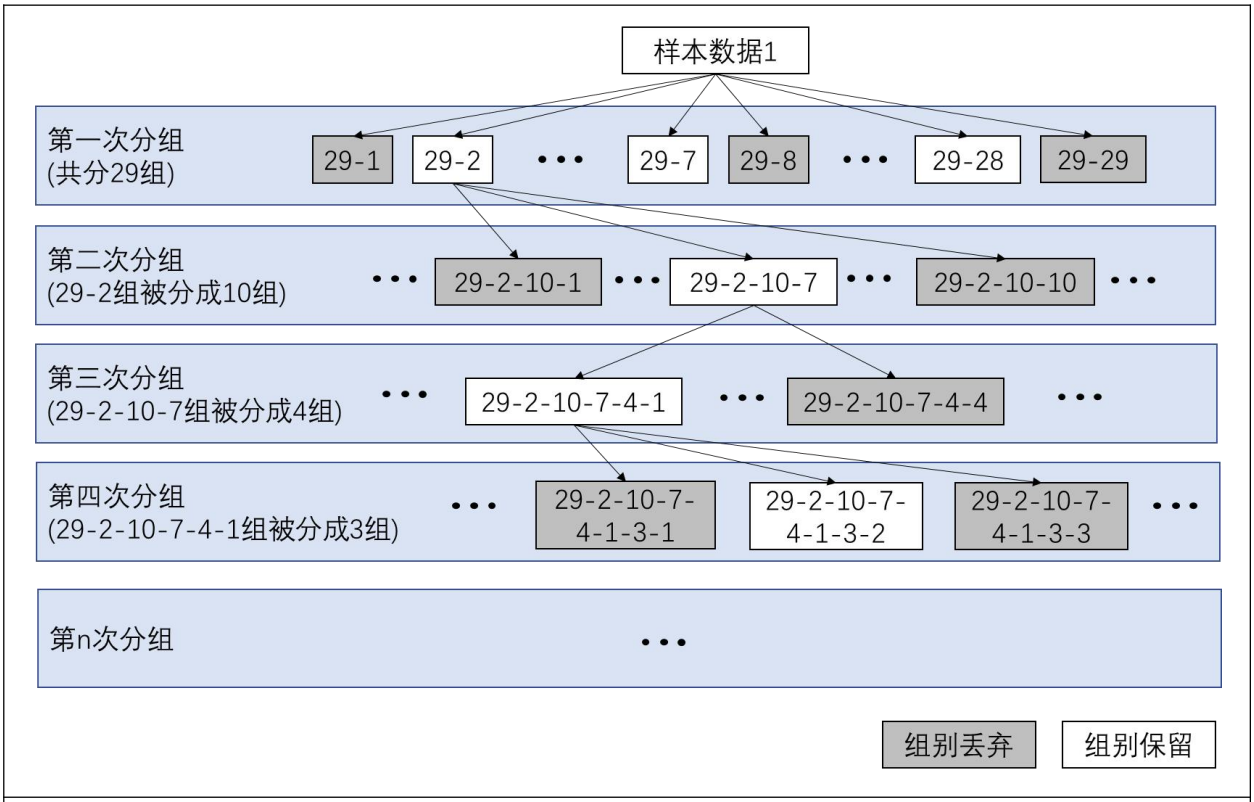

图9 组别演化过程

图中灰色方框代表本组被丢弃，白色方框代表本组被保留，例如图中第一次分组后，29-2组与化合物生物活性相关性强，该组被保留，由于组内相关性低于阈值，29-2组参与第二次分组；同理，29-2-10-7参与第三次分组，29-2-10-7-4-1参与第四次分组。29-2-10-7-4-1-3-2组与化合物生物活性相关性强，且组内相关性低于阈值，该组被保留且无需再分组，将直接参与后续环节。

3.3.3.3 结果

将729个分子按照相关性分组，选取10组为例，如表3所示，表内展示了排序顺序、组别、组内分子变量。

表 3 最终分类结果（选 10 组为例）

排序	组别	组内分子变量
1	29-16-4-4-3-3-2-2-2-2	LipoaffinityIndex
		MLogP
		MDEC-23
2	29-2-20-7-4-3-4-1-3-2	C1SP2
3	29-7-6-5-2-1-2-1-2-2-2-2-1	nC
		ATSp2
		ATSp5
		VP-3
4	29-21	minHBd
		minHsOH
		minsOH
		maxHBd
		maxHsOH
		maxsOH
5	29-23-5-5-3-2-2-2	CrippenLogP
6	29-26-8-8-2-1	AMR
		ATSm3
		ATSm4
		ATSm5
		ATSp1
		SC-3
		SP-3
		ETA_Eta_B_RC
		WPOL
		Zagreb
7	29-26-8-1-3-3-2-1	apol
		nAtom
		nH
		nBonds2
		nBondsS
		nBondsS2
		nBondsS3
		bpol
		C2SP3
		VP-1
		VP-2
		nHCsats

		ETA_Eta_L
8	29-12-5-5-2-2	hmin
9	29-2-20-8-6-4-4-3	nBase
		fragC
10	29-26-8-1-3-1-2-1-2-1	nHeavyAtom
		ATSm1
		ATSm2
		nBonds
		SP-0
		SP-1
		SP-2
		VP-0
		CrippenMR
		ETA_Alpha
		ETA_Beta_s
		ETA_Eta_R_L
		Kier1
		McGowan_Volume
		MLFER_L
		VABC
		MW
		WTPT-1
		MLFER_BH
		MLFER_BO

3.3.4模型三求解过程

3.3.4.1算法

模型三算法如图10所示。

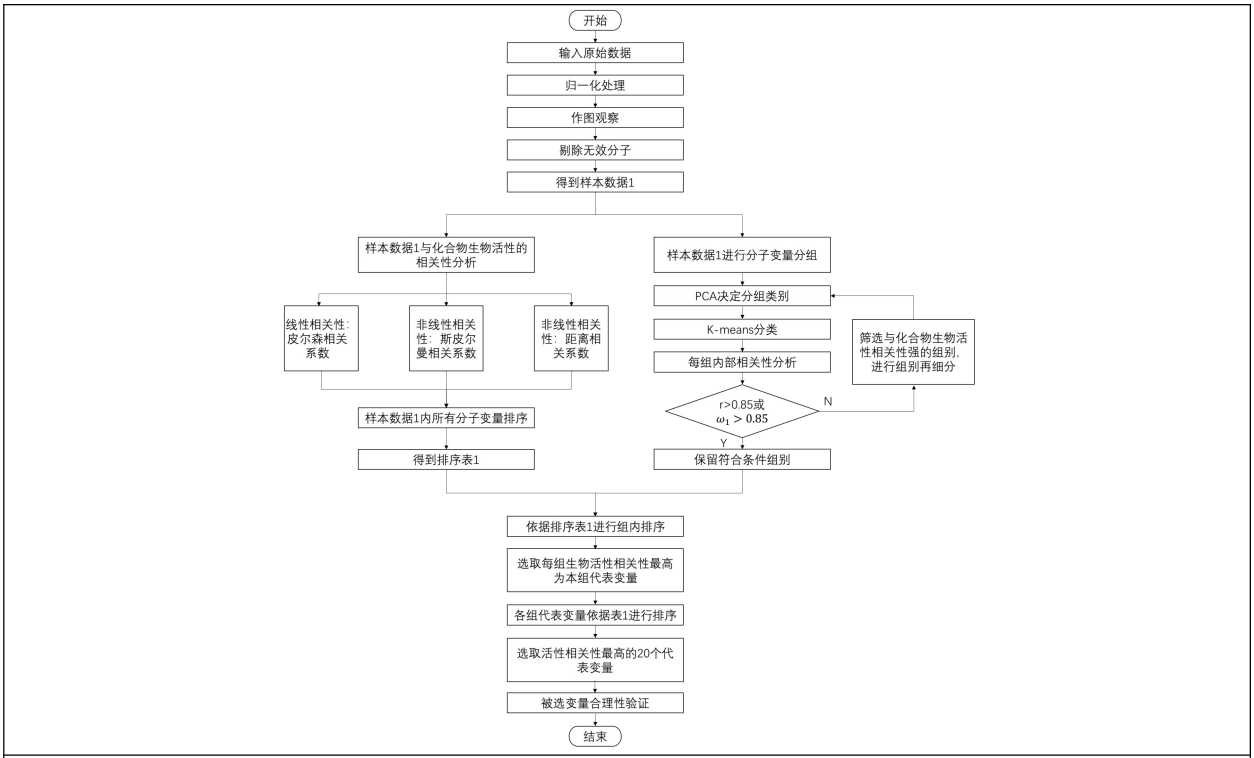

图10 基于相关性指标与聚类结果的变量选择模型算法

步骤一：归一化处理，去掉量纲对数据的影响，将数据集中于区间$(0，1)$。

步骤二：作图观察离散点的分布，初步估计相关性。

步骤三：剔除原数据集中的无关变量，减少干扰项，得到样本数据1。

步骤四：对样本数据1进行活性相关性分析，采用三种方法分析，综合结果分析。

步骤五：根据相关性分析结果，对分子变量进行排序

步骤六：验证排序结果。

3.3.3.2结果

依照排序表，剔除化合物生物活性低的组别，每组选出相关性系数最高的分子变量作为代表，所有分子变量代表进行排序，选取前20个分子变量，最终的分类结果如表4所示。

表4 被选取的20个分子变量

排序	分子名称	相关性综合评价模型结果
1	MDEC-23	0.549051863443223
2	C1SP2	0.502389662574605
3	nC	0.499132689824706
4	maxsOH	0.480184605885039
5	CrippenLogP	0.473745741842321
6	AMR	0.471977307561249
7	apol	0.462082368446228
8	hmin	0.458561862206639
9	fragC	0.453174089573062
10	VABC	0.449463366895657
11	SwHBa	0.447597960234714

12	SP-5	0.446965152390801
13	C2SP2	0.446227433715045
14	nT6Ring	0.440443217304941
15	minsssN	0.438910342947204
16	MDEC-22	0.437949883767128
17	VP-5	0.435015426625137
18	BCUTp-1h	0.432878249679374
19	SsOH	0.43226707040393
20	ETA_Eta_R	0.42874061729812

依据表4选取MDEC-23、CISP2、minsssN、hmin四个分子作为自变量，pIC50作为因变量作图，如图11所示。

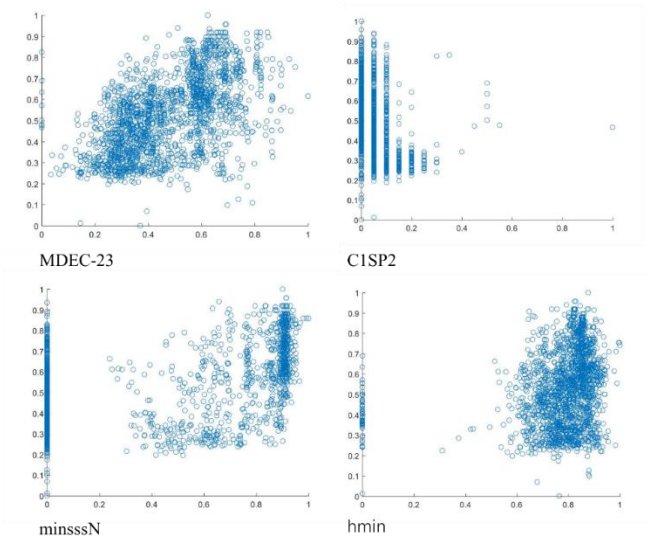

图11部分分子变量展示

4 问题二的建模与求解.

4.1 问题分析

问题一采用三个模型，进行了相关性分析排序、分子自变量分组、选出各组分子自变量代表、对分子自变量代表进行排序、选出前20个分子自变量。本题结合问题一的分子自变量，构建化合物对ERα生物活性的定量预测模型，对文件“ERα_activity.xlsx”的test表中的50个化合物进行IC50值和对应的pIC50值预测。

可以将问题简化成一个曲线拟合预测问题，有以下2个难点：

（1） 非线性拟合算法的选取；

（2） 模型参数调试；

针对以上两个难点，做如下处理：

（1） 对自变量分别做线性拟合和非线性拟合，初步观察拟合情况，结合分子自变量特点选取拟合算法；

（2） 利用网格调参法调节模型参数，选取最优情况；

4.2 基于adaBoost的预测模型建立

问题二为一预测问题，其目的是在现有分子描述符值及其对应生物活性值的基础上，通过挖掘两者的信息，在得到未知化合物的分子描述符信息后，预测该化合物对ERα的生

物活性值IC_{50}、pIC_{50}，并要求精度尽可能高。该问题的核心在于挖掘分子描述符值及生物活性值之间的关系，建立关于分子描述符值的生物活性值拟合回归预测模型，具体如下所示。

目标模型：

$$Y_j = g(X_1''', X_2''', ..., X_q'''; \tilde{\theta}_1, \tilde{\theta}_2, \cdots \tilde{\theta}_p) \tag{21}$$

约束条件：

（1）由题目要求：$1 \le q \le 20$；

（2）$\left\|Y_j - g\left(X_1''', X_2''', ..., X_q'''\right)\right\|^2 = \min\limits_{h(\cdot)\in H} \left\|Y_j - h\left(X_1''', X_2''', ..., X_q'''\right)\right\|^2$；

（3）$\left\|Y_j - g\left(X_1''', ..., X_q'''; \tilde{\theta}_1, \tilde{\theta}_2, \cdots \tilde{\theta}_p\right)\right\|^2 = \min\limits_{\theta_1, \theta_2, \cdots \theta_p \in \boldsymbol{\theta}} \left\|Y_j - g\left(X_1''', ..., X_q'''; \theta_1, \theta_2, \cdots \theta_p\right)\right\|^2$，其中，

$\boldsymbol{\theta}$为所有参数集合:$\theta_1, \theta_2, \cdots \theta_p$

4.3求解过程

4.3.1算法

首先从文件“Molecular_Descriptor. Xlsx”的训练集中提取问题一中的20个分自变量，对其进行归一化处理，再进行PCA降维，降维后的数据按照0.2的比例分出测试集，余下皆是训练集数据对训练网络对网络行优化，将模型存储，模型调试算法，如图12所示。

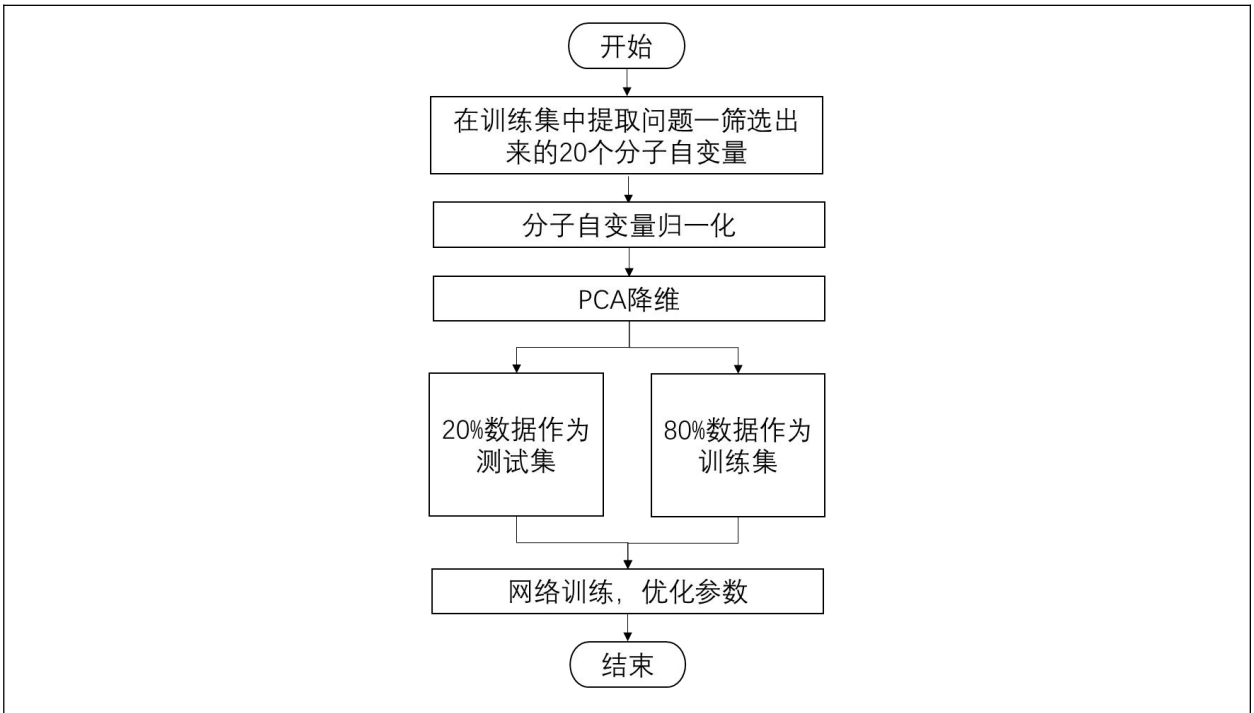

图12 模型调试算法

在预测部分，首先加载已经训练好的网络，从文件“ERα_activity.xlsx”测试集中筛选出问题一的20个分子变量，分子便变量归一化后PCA降维，输入已近训练好的网络，后进行反归一化预测pIC50的值，如图13所示。

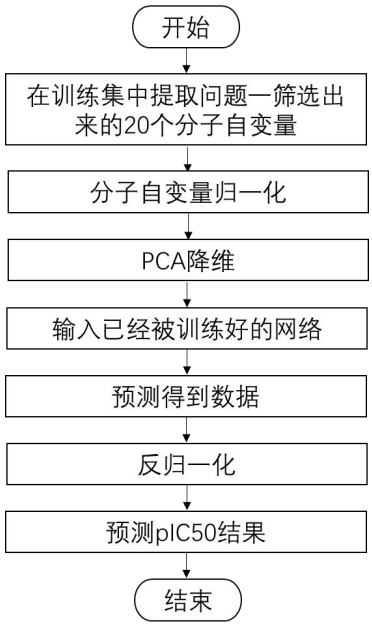

图13 模型预测算法

4.3.2求解

模型调试完成，用图12中的测试集测试模型，如图14所示

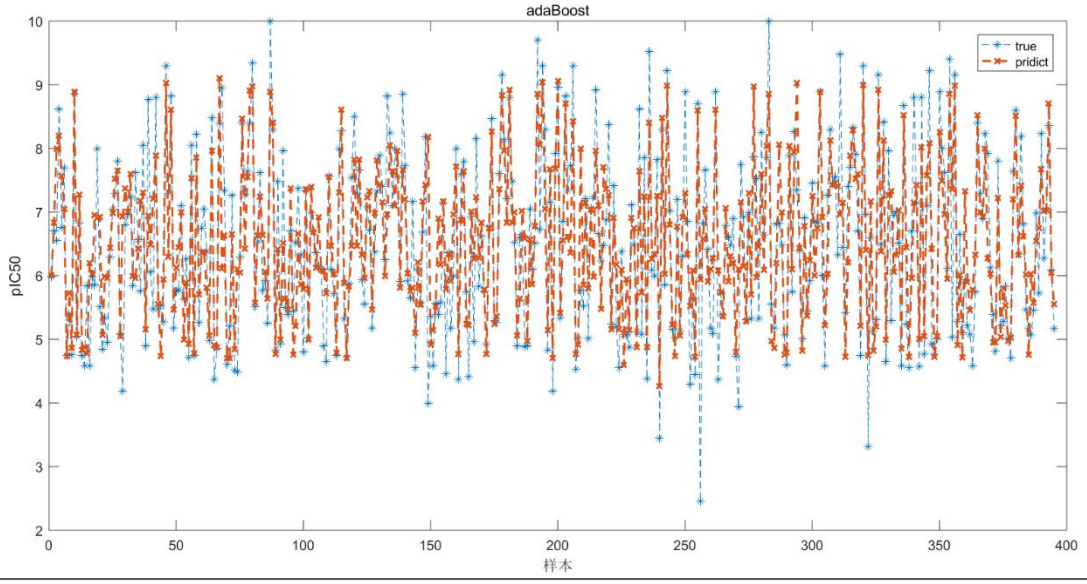

图14 模型调试结果

图中蓝色点为真实值，红色点为预测值。

决定系数R^2的定义为：

$$R^2 = 1 - \frac{\|Y_{true} - Y_{pred}\|^2}{\|Y_{true} - \overline{Y}_{true}\|^2} \quad (22)$$

针对该模型：

$$R^2 = 0.7022053011531851$$

$$MAE = 0.55611$$

4.3.3结果

展示模型在图15上的测试值的结果如图15所示，图中展示了部分结果。

SMILES	IC50_nM	pIC50
c2ccc(OS(=O)(=O)[C@@H]3C[C@@H]4O[C@H]3C(=C4c5ccc(O)cc	10997	4.958
OC(=O)\C=C\c1ccc(cc1)C2=C(CCOc3ccccc23)c4ccc(O)cc4	1895	5.721
COc1ccc2C(=C(CCOc2c1)c3ccc(O)cc3)c4ccc(\C=C\C(=O)O)cc4	2262	5.644
OC(=O)\C=C\c1ccc(cc1)C2=C(CCOc3cc(F)ccc23)c4ccc(O)cc4	4171	5.379
OC(=O)\C=C\c1ccc(cc1)C2=C(CCSc3cc(F)ccc23)c4ccc(O)cc4	16888	4.772
CC(=O)\C=C\c1ccc(cc1)C2=C(CCOc3cc(F)ccc23)c4ccc(O)cc4	2524	5.597
Oc1ccc(cc1)C2=C(c3ccc(\C=C\c4ccccc4)cc3)c5ccc(F)cc5OCC2	2424	5.614
Oc1ccc(cc1)C2=C(c3ccc(\C=C\C(=O)c4ccccc4)cc3)c5ccc(F)cc5OCC2	4103	5.386
OC(=O)\C=C\C=C\c1ccc(cc1)C2=C(CCOc3cc(F)ccc23)c4ccc(O)cc4	7077	5.149
CCN(CC)C(=O)\C=C\c1ccc(cc1)C2=C(CCOc3cc(F)ccc23)c4ccc(O)cc4	883	6.053
Oc1ccc(cc1)C2=C(c3ccc(\C=C\C(=O)N4CCCC4)cc3)c5ccc(F)cc5OCC2	2150	5.667
N(CC)CCNC(=O)\C=C\c1ccc(cc1)C2=C(CCOc3cc(F)ccc23)c4ccc(O)c	625	6.203
c1ccc(cc1)C2=C(c3ccc(\C=C\C(=O)N4CCNCC4)cc3)c5ccc(F)cc5OCC	2047	5.688
1CCN(CC1)C(=O)\C=C\c2ccc(cc2)C3=C(CCOc4cc(F)ccc34)c5ccc(O)	1680	5.774
cc1)C2=C(c3ccc(\C=C\C(=O)N4CCN(Cc5ccccc5)CC4)cc3)c6ccc(F)c	2893	5.538

图15 预测数据

模型预测出的值是pIC50，pIC50和IC50是负对数关系，如图16为两值之间关系式。

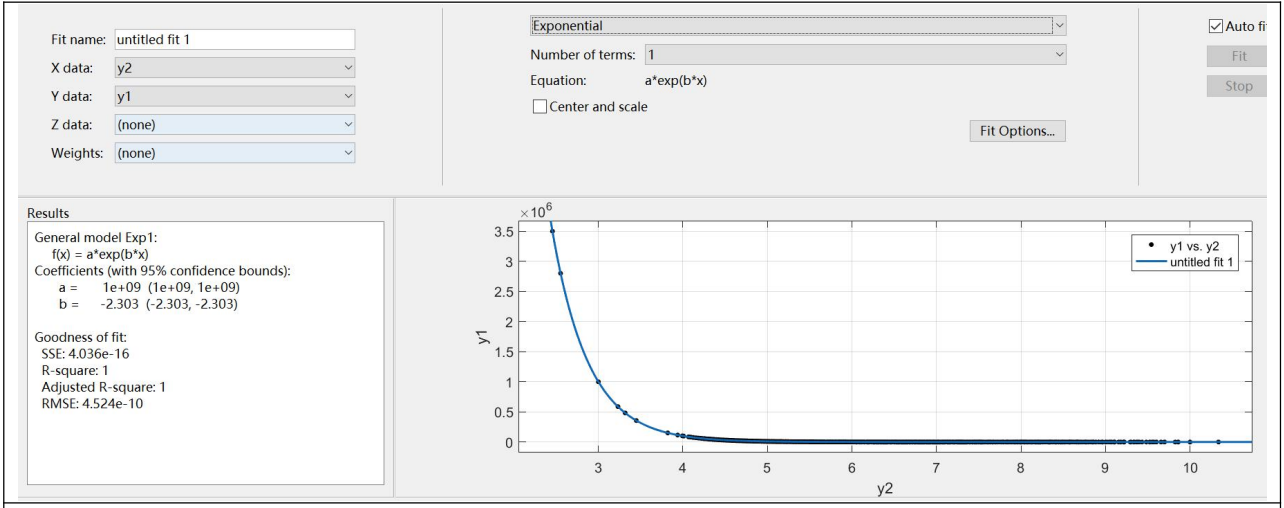

图16 活性关系

5 问题三的建模与求解

5.1 问题分析

优化药物成分的时候需要考虑化合物的ADMET性质，题目从5个角度描述化合物的ADMET性质，分别是Caco-2、CYP3A4、hERG、HOB、MN，要求根据提供的1974个化合物的分子信息和ADMET信息描述进行分类预测。

该问题可以被简化成针对ADMET的五个角度分别建立二分类算法。该问题具有以下3个难点：

（1） 分子自变量的取舍；

（2） 小数据集的二分类算法；

（3） 模型参数的调试；

针对这三个难点，做如下处理：

（1） 第一题将分子自变量进行PCA主成分分析，可知分子自变量内蕴含的特征有29个，经过第一题的相关性综合评价模型、基于PCA与K-means算法的分子聚类模型与基于相关性指标与聚类结果的变量选择模型挑选的20个分子在29个特征中的具有代表性，故而无需再挑选；

（2） 分析常见二分类模型特征，并做一定的验证，最终选定合适算法；

（3） 采用网格调参法进行模型调参，得到理想模型。

5.2 基于PCA-SVC-KNN的ADMET预测模型建立

问题三为5个二分类问题，其目的是在现有分子描述符值及其对应的ADMET性质的基础上，挖掘其中信息，旨在得到未知化合物的分子描述符信息后，依次预判该化合物的Caco-2、CYP3A4、hERG、HOB、MN值，并要求准确率尽可能高。该问题的核心在于挖掘分子描述符值与ADMET性质之间的关系，基于分子描述符依次建立Caco-2预判模型、CYP3A4预判模型、hERG预判模型、HOB预判模型和MN预判模型。如下所示：

目标模型：

$$Z_i = \hbar_i(X_1^{''}, X_2^{''}, \ldots, X_{q_i}^{''}; \widetilde{\varphi}_{i1}, \widetilde{\varphi}_{i2}, \cdots \widetilde{\varphi}_{ip_i}) \tag{23}$$

约束条件：

（1）$1 \le q_i \le 729$，$i = 1,2,\cdots,5$；

（2）$\left\|Z_i - \hbar_i(X_1^{''}, X_2^{''}, \ldots, X_{q_i}^{''})\right\|^2 = \min\limits_{\hbar(\cdot)\in H_1} \left\|Z_i - h\left(X_1^{'''}, X_2^{'''}, \ldots, X_q^{'''}\right)\right\|^2$，$i = 1,2,\cdots,5$；；

（3）$\left\|Z_i - \hbar_i(X_1^{''}, \ldots, X_{q_i}^{''}; \widetilde{\varphi}_{i1}, \cdots \widetilde{\varphi}_{ip_i})\right\|^2 = \min\limits_{\varphi_{i1},\cdots\varphi_{i1_{ip_i}}\in\boldsymbol{\varphi}} \left\|Z_i - \hbar_i\left(X_1^{'''}, \ldots, X_{qq_i}^{''}; \varphi_{i1}, \cdots \varphi_{i1_{ip_i}}\right)\right\|^2$，

其中，$\boldsymbol{\varphi}$该模型为所有参数集合。

5.2.1 基础理论

1）SVC分类算法：

给定二分类训练向量$x_i \epsilon R^n$，$i = 1, \ldots, l$,给定一个目标向量$y \epsilon R^l$例如$y_i \epsilon \{1, -1\}$,C-SVC用于以下原始优化问题：

$$\begin{gathered} \min\limits_{\omega,\ b,\xi} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}\xi_i \\ y_i(\omega^T\phi(x_i) + b) \ge 1 - \xi_i \\ \xi_i \ge 0, i = 1, \ldots, l, \end{gathered} \tag{24}$$

公式中$\phi(x_i)$将x_i投影到更高维空间并且$C > 0$是正则化参数。由于变量向量ω有高纬度的可能性，通常我们要解决另外一个问题：

$$\begin{gathered} \min\limits_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \\ y^T\alpha = 0 \\ 0 \le \alpha_i \le C, i = 1, \ldots, l, \end{gathered} \tag{25}$$

其中$e = [1, \ldots, 1]^T$,Q是$l * l$的半正定矩阵，$Q_{ij} \equiv y_i y_j K(x_i, x_j)$,并且$K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$

是核函数。
在以上问题解决后，采用原始对偶关系，最佳的ω满足：

$$\omega = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i) \tag{26}$$

决策函数为：

$$sgn(\omega^T \phi(x_i) + b) = sgn\left(\sum_{i=1}^{l} y_i \alpha_j \phi(x_i, x_j) + b\right) \tag{27}$$

其中$y_i \alpha_j \ \forall i, b$,标签名字，和其他信息例如模型中的核函数被用于预测[6]。

2）KNN分类算法：
KNN主要公式为：

$$L(x_i, x_j) = \left(\sum_{l=1}^{n} \left|x_i^{(l)} - x_i^{(l)}\right|^2\right)^{\frac{1}{2}} \tag{28}$$

将预测点与所有点距离进行计算，将距离排序，选取最靠近预测点的k个点，这些点中的多数样本决定预测点的标签。
5.3模型求解
5. 3.1算法
针对ADMET的5个性质，采用SVC和KNN两种算法
1）模型一：基于PCA与SVC的ADMET预测模型
该模型将用于ADMET中的CYP3A4、hERG、MN三种性质的分类和预测，模型算法如图17所示。

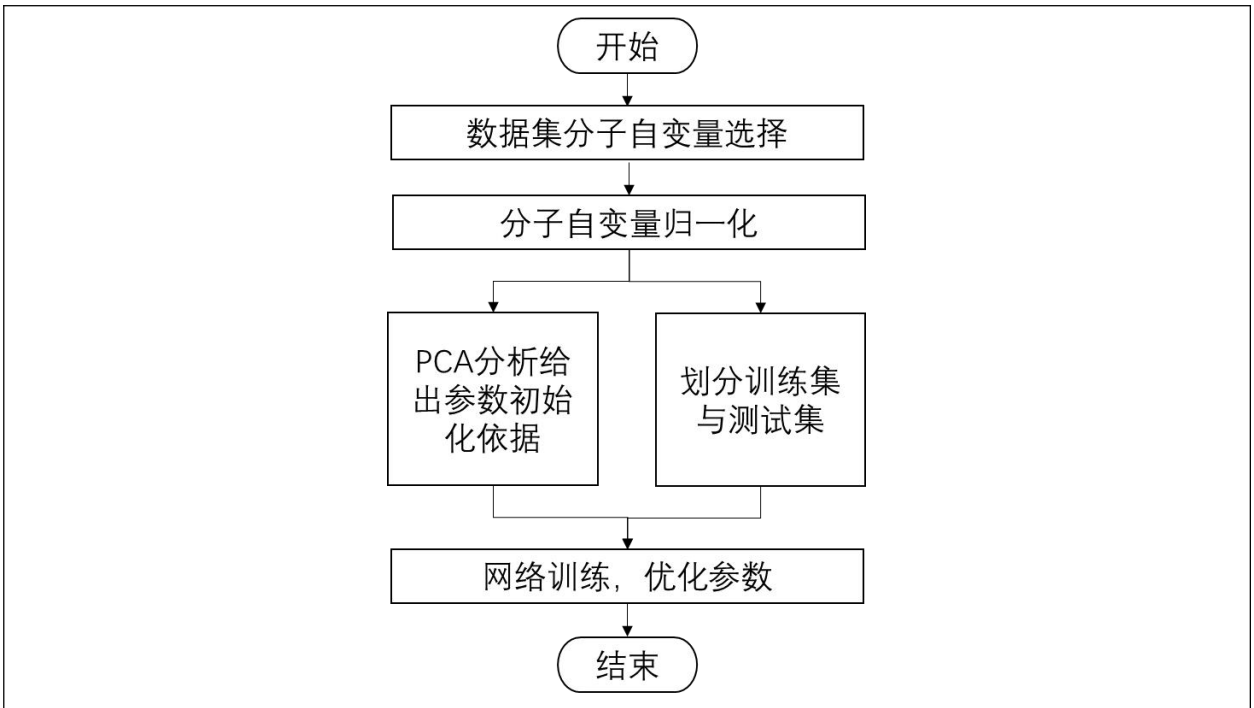

图17 基于PCA与SVC的ADMET预测模型算法

首先进行数据集分子自变量选择，减少重复信息；再分子自变量归一化，消除量纲对分子变量的影响；依照PCA分析结果初始化网络以及按照0.2的比率在数据集中划分数据集；最后网络训练，实现参数优化。

2）模型二：基于PCA与KNN的ADMET预测模型

该模型将用于ADMET中的CaCo-2、HOB两种性质的分类和预测，模型算法如图18所示。

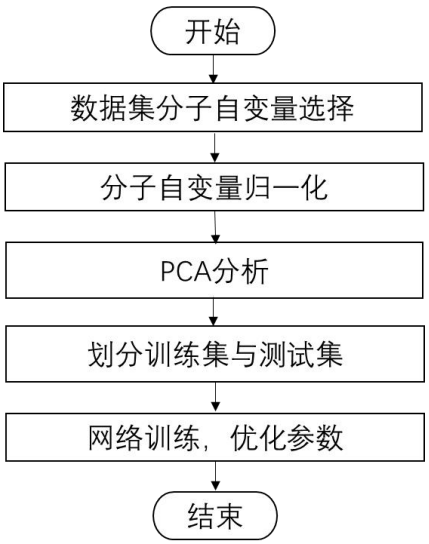

图18 基于PCA与KNN的ADMET预测模型算法

首先进行数据集分子自变量选择，减少重复信息；再分子自变量归一化，消除量纲对分子变量的影响；依照PCA分析结果在数据集中划分数据集；最后网络训练，实现参数优化。

5.3.2求解

1）模型一求解过程

将样本数据集按照0.2的比例划分成训练集和测试集，经过训练，模型的准确率以及各参数设置如表5所示。

表5 基于PCA与SVC的ADMET预测模型参数与准确率

ADMET	算法	网络超参数设置				PCA降维	准确率
		C	kernel	gamma	decision_function_shape		
CYP3A4	SVC	10	‘rbf’	16	‘ovr’	12	93.92
hERG	SVC	1	‘rbf’	30	‘ovr’	9	88.35%
MN	SVC	40	‘rbf’	14	‘ovr’	8	94.94%

图19为CYP3A4、hERG、MN三性质的模糊矩阵。

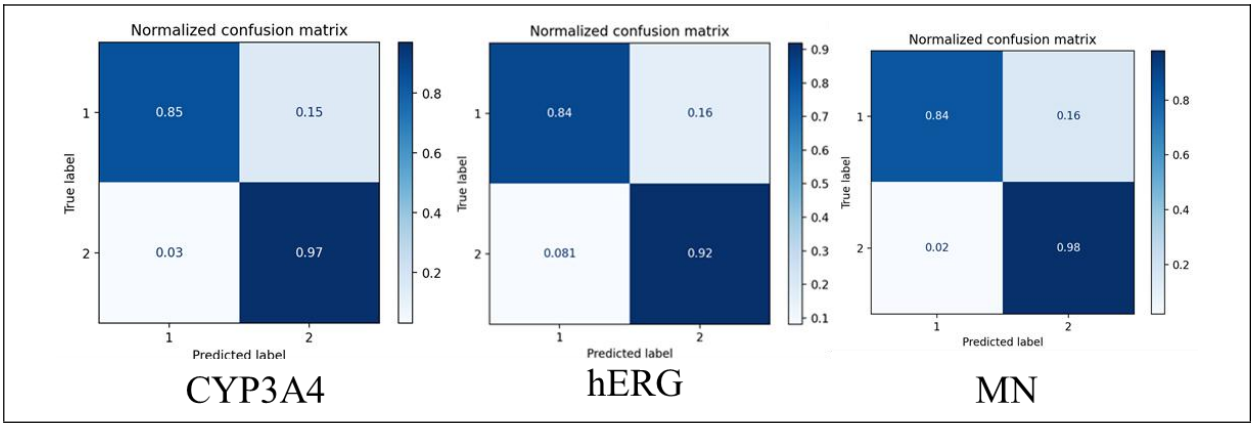

图19 基于PCA与SVC的ADMET预测模型的模糊矩阵

2）模型二求解过程

基于PCA与KNN的ADMET预测模型经过训练，模型的准确率以及各参数设置如表6所示。

表6 基于PCA与KNN的ADMET预测模型参数与准确率

ADMET	算法	k	PCA降维	准确率
CaCo-2	KNN	80	5	100%
HOB	KNN	100	8	100%

由于KNN是基于数据集的算法，需要在原数据集中抽取一定数量的样本用于训练，图20是训练样本数和预测准确率的关系，CaCo-2性质中样本数量在400到500之间预测准确率开始急剧下降，HOB性质中样本数量在800到900之间开始急剧下降。

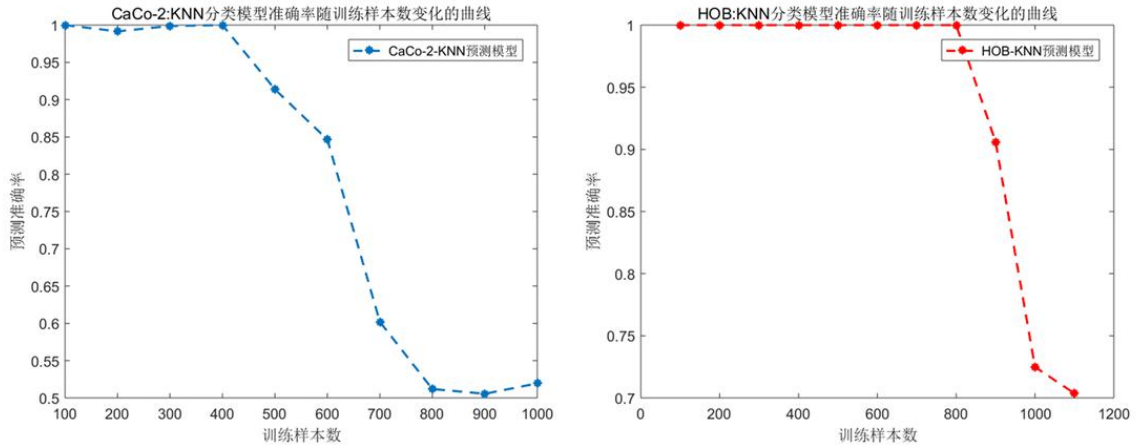

图20 基于PCA与KNN的ADMET预测模型的样本数-预测准确率关系图

5．3．3结果

如图21所示，为模型一和模型二预测结果的汇总表，图中选取前15个截图示意。

SMILES	Caco-2	CYP3A4	hERG	HOB	MN
=C\c2ccc(OS(=O)(=O)[C@@H]3C[C@@H]4O[C@H]3C(=C4c5ccc(O)cc5)c6	0	1	1	0	1
OC(=O)\C=C\c1ccc(cc1)C2=C(CCOc3ccccc23)c4ccc(O)cc4	0	1	1	0	1
COc1ccc2C(=C(CCOc2c1)c3ccc(O)cc3)c4ccc(\C=C\C(=O)O)cc4	0	1	1	0	1
OC(=O)\C=C\c1ccc(cc1)C2=C(CCOc3cc(F)ccc23)c4ccc(O)cc4	0	1	1	0	1
OC(=O)\C=C\c1ccc(cc1)C2=C(CCSc3cc(F)ccc23)c4ccc(O)cc4	0	1	1	0	1
CC(=O)\C=C\c1ccc(cc1)C2=C(CCOc3cc(F)ccc23)c4ccc(O)cc4	0	1	1	0	1
Oc1ccc(cc1)C2=C(c3ccc(\C=C\c4ccccc4)cc3)c5ccc(F)cc5OCC2	0	1	1	0	1
Oc1ccc(cc1)C2=C(c3ccc(\C=C\C(=O)c4ccccc4)cc3)c5ccc(F)cc5OCC2	0	1	1	0	1
OC(=O)\C=C\C=C\c1ccc(cc1)C2=C(CCOc3cc(F)ccc23)c4ccc(O)cc4	0	1	1	0	1
CCN(CC)C(=O)\C=C\c1ccc(cc1)C2=C(CCOc3cc(F)ccc23)c4ccc(O)cc4	0	0	0	0	0
Oc1ccc(cc1)C2=C(c3ccc(\C=C\C(=O)N4CCCC4)cc3)c5ccc(F)cc5OCC2	0	0	1	0	0
CCN(CC)CCNC(=O)\C=C\c1ccc(cc1)C2=C(CCOc3cc(F)ccc23)c4ccc(O)cc4	0	0	1	0	1
Oc1ccc(cc1)C2=C(c3ccc(\C=C\C(=O)N4CCNCC4)cc3)c5ccc(F)cc5OCC2	0	0	1	0	1
CN1CCN(CC1)C(=O)\C=C\c2ccc(cc2)C3=C(CCOc4cc(F)ccc34)c5ccc(O)cc5	0	0	1	0	1
ccc(cc1)C2=C(c3ccc(\C=C\C(=O)N4CCN(Cc5ccccc5)CC4)cc3)c6ccc(F)cc6O	0	1	1	0	1

图21 ADMET的部分预测结果截图

6 问题四的建模与求解

6.1 问题分析

题目要求寻找兼顾生物活性和ADMET性质的分子变量。这可以看作一个多约束的统计问题，主要有以下难点，

（1）化合物的分子描述符数量过多，需要在数量繁多的分子描述符中定量分析分子描述符取值对ERα生物活性的影响。

（2）既要考虑化合物的生物活性，又要兼顾化合物的ADMET性质。

（3）多目标优化算法复杂性较大。

针对这三个难点，做如下处理：

（1） 针对分子描述符多的问题，经过问题一的分析和验证，筛选出了20个和ERα生物活性相关性最高且相互之间相关性较低的分子描述符。因此，对ERα生物活性影响最大的分子描述符均在筛选出的20个变量中，其它相关性较小的分子描述符忽略，仅仅对筛选出的20个分子描述符进行分析。

（2） 问题约束较多的问题，对训练数据集进行统计分析，最后建立基于统计分析和假设检验的模型，并统计出分子描述符对ERα生物活性和ADMET性质定量分析的模型。该模型简单，克服多目标优化算法复杂度大的问题。

6.2 基于K-S检验的分子筛选与取值范围的模型建立

问题四为多目标条件优化问题，旨在结合问题二对生物活性的预测模型与问题三对于ADMET性质的判别模型，筛选分子描述符，并构建多分子描述符取值优化模型，探索在当生物活性预测值尽可能大，且ADMET性质预判值至少有三个较好时，分子描述符值的取值范围。构建的模型如下：

目标模型：

$$\max \ ①$$

$$\max \ ②$$

约束条件：

（1）$① = g(X_1''', X_2''', ..., X_q''')$；

（2）$② = \sum_{i=1,2,4} \hbar_i(X_1'', X_2'', ..., X_{q_i}'') + \sum_{i=3,5} h_0\left[\hbar_i(X_1'', X_2'', ..., X_{q_i}'')\right]$，其中$h_0(\cdot)$为转换函数，计算公式为：

$$h_0(x) = |x - 1|$$

（3）$② \geq 3$；

（4）$m_i''' \leq X_i''' \leq M_i'''$，$1 \leq i \leq q$；

（5）$m_i'' \leq X_i'' \leq M_i''$；

（1）$1 \leq q_i \leq 729$，$i = 1,2,\cdots,5$；

（2）$\left\|Z_i - \hbar_i(X_1'', X_2'', ..., X_{q_i}'')\right\|^2 = \min_{\hbar(\cdot)\in H_1} \left\|Z_i - h\left(X_1''', X_2''', ..., X_q'''\right)\right\|^2$，$i = 1,2,\cdots,5$；；

（3）$\left\|Z_i - \hbar_i(X_1'', ..., X_{q_i}''; \widetilde{\varphi}_{i1}, \cdots \widetilde{\varphi}_{ip_i})\right\|^2 = \min_{\varphi_{i1},\cdots\varphi_{i1_{ip_i}}\in\boldsymbol{\varphi}} \left\|Z_i - \hbar_i\left(X_1''', ..., X_{qq_i}''; \varphi_{i1}, \cdots \varphi_{i1_{ip_i}}\right)\right\|^2$，

其中，$\boldsymbol{\varphi}$该模型为所有参数集合。

6.3求解过程

6.3.1算法

该模型的算法求解过程如图22。

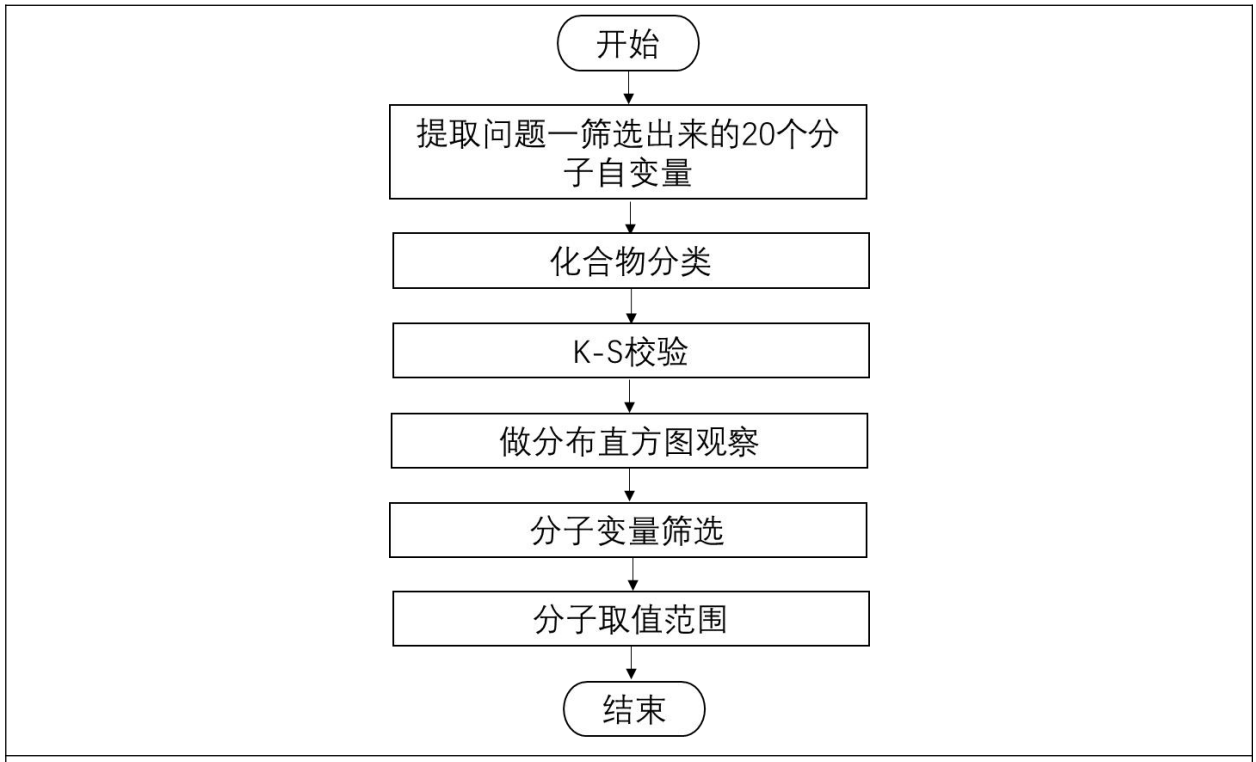

图22 基于K-S检验的分子筛选与取值范围的模型算法

步骤一：提取问题一中的20个分子变量；

步骤二：对化合物进行分类，分类要求是依照pIC活性排序，满足排在前25%的分子同时ADMET性质至少三个性质较好的要求为第一类，余下的为第二类；

步骤三：两类化合物从分子角度进行K-S检验，初步筛选去除部分分子；

步骤四：余下的分子做分布直方图观察，去除差异性小的分子；

步骤五：确定分子范围；

6.3.2求解

经过K-S检验和直方图分布观察，筛选结果如表7所示。黄色代表没有通过K-S检验的分子，绿色代表直方图检查未通过分子。

表7 筛除结果

排序	分子名称	相关性综合评价模型结果
1	MDEC-23	0.549051863443223
2	C1SP2	0.502389662574605
3	nC	0.499132689824706
4	maxsOH	0.480184605885039
5	CrippenLogP	0.473745741842321
6	AMR	0.471977307561249
7	apol	0.462082368446228
8	hmin	0.458561862206639
9	fragC	0.453174089573062
10	VABC	0.449463366895657
11	SwHBa	0.447597960234714
12	SP-5	0.446965152390801
13	C2SP2	0.446227433715045

14	nT6Ring	0.440443217304941
15	minsssN	0.438910342947204
16	MDEC-22	0.437949883767128
17	VP-5	0.435015426625137
18	BCUTp-1h	0.432878249679374
19	SsOH	0.43226707040393
20	ETA_Eta_R	0.42874061729812

选取MDEC-23、BCUTp-1h、SsOH三个分子作图观察，如图23所示

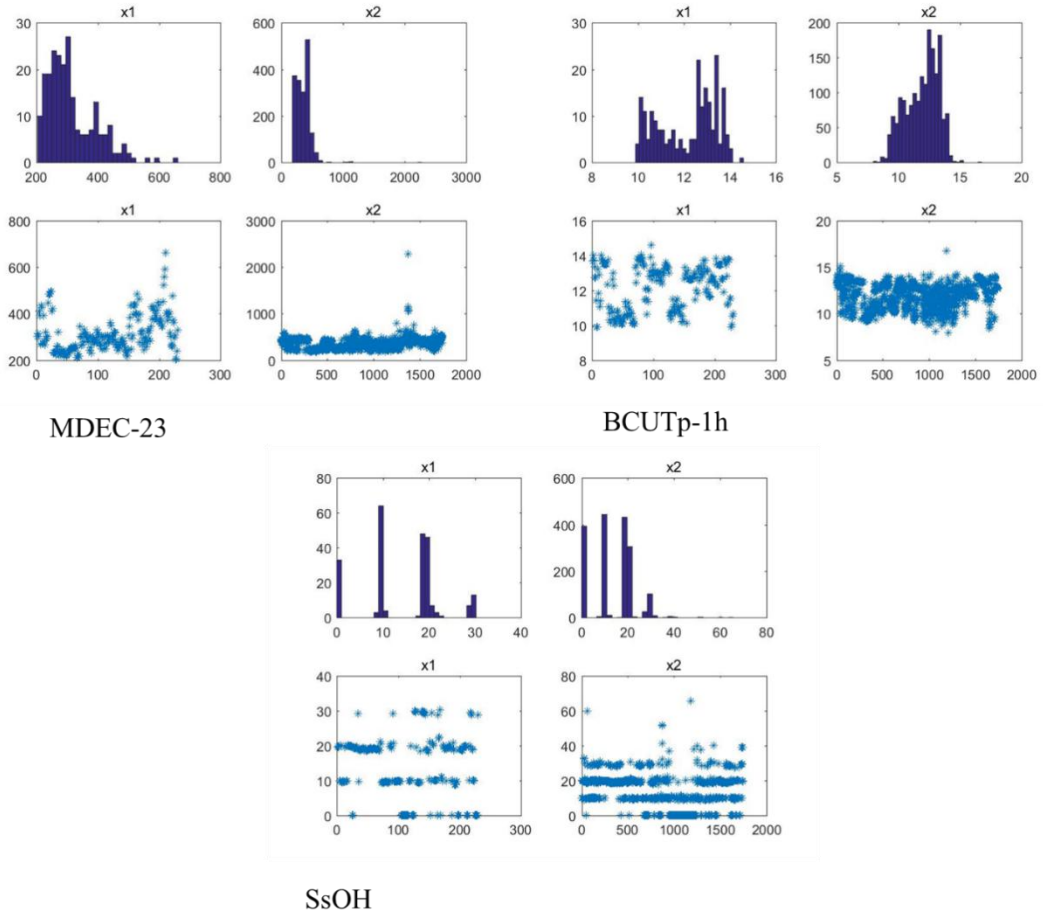

图23 部分分子直方图

剔除不需要的分子后，表8为筛除结果

表8 筛除结果

排序	分子名称	相关性综合评价模型结果
1	MDEC-23	0.549051863443223
2	C1SP2	0.502389662574605
3	nC	0.499132689824706
4	AMR	0.471977307561249
5	apol	0.462082368446228
6	hmin	0.458561862206639
7	fragC	0.453174089573062
8	VABC	0.449463366895657
9	SwHBa	0.447597960234714
10	SP-5	0.446965152390801
11	C2SP2	0.446227433715045
12	MDEC-22	0.437949883767128
13	VP-5	0.435015426625137
14	ETA_Eta_R	0.42874061729812

6.3.3结果

对筛除结果进行取值范围计算，计算结果如表9所示。

表9 取值范围

排序	分子名称	取值范围
1	MDEC-23	[0，22.70]
2	C1SP2	0
3	nC	[7，20]
4	AMR	[54.07，103.17]
5	apol	[30.66，54.09]
6	hmin	[-0.0352，-0.5888]
7	fragC	[274.06，1729.44]
8	VABC	[182.55，309.42]
9	SwHBa	[-21.76，19.28]
10	SP-5	[1.85，6.754]
11	C2SP2	[0，11]
12	MDEC-22	[0，11.72]
13	VP-5	[0.61，2.77]
14	ETA_Eta_R	[18.68，44.26]

7 模型的评价与改进方向

1.1 模型的优点：

1）充分考虑了变量之间可能存在的相关性或者非相关性，又考虑到了变量间的耦合性，使用的数据处理和变量筛选模型合理，类型变量筛选均可采用，能得到高相关和低耦合的数据变量。

2）经过比较和选择，建立的adaBoost预测模型较常用的随机森林、决策树、BP神经网络具有更好的预测效果，在测试集上具有较好的预测精度和鲁棒性。

3）针对数据量较小的问题，选择了在小样本集上表现优秀的KNN和SVM算法为分类模型基础，建立的模型具有较高的预测精度，鲁棒性也较好。

4）最后的搜索和优化模型，采用了统计和显著性检验为基础建立，相对最优化算法复杂度低，通用性也较强。

5）算法复杂度低，响应块。

1.2 模型的缺点:

1）处理时认为所有操作变量均独立可变，与实际可能不符合。

2）NN算法对训练样本较小时表现比较好，如果训练样本数据较多，表现会下降,因此，训练样本数不宜过多。

3).最后统计分析模型只考虑了筛选出的操作变量，其它的变量有部分也可能对活性造成影响。

1.3 模型的改进:

1）分类预测模型中可以分析分子描述符和ADMET性质之间的关系，进行综合评判后再进行分类，有可能进一步提高模型效果。

2）KNN分类模型的训练样本数的确定要较为合理，并不适合把所有的训练数据用于训练。

3）对于问题4的优化问题，还可以进一步分析其它在第一问中筛选掉的某些相关性大的分子描述符。

参考文献

[1]陈志方.中国货币政策的有效性评估——基于皮尔森相关系数的分析[J].中国商论,2020(06):48-49.
[2] Székely G J, Rizzo M L, Bakirov N K. Measuring and testing dependence by correlation of distances[C]. Acm Symposium on Virtual Reality Software & Technology, 2007.
[3]张新生,蔡宝泉.基于改进随机森林模型的海底管道腐蚀预测[J].中国安全科学学报,2021,31(08):69-74.
[4]Microstrong,主成分分析(PCA)原理详解,https://zhuanlan.zhihu.com/p/37777074,2021年10月16日。
[5] 饕餮纹, 分类模型——k-means, https://zhuanlan.zhihu.com/p/158036185, 2021年10月17日
[6] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifers. In Proceedings of the Fifth Annual Workshop on Computational LearningTheory, 144-152. ACM Press, 1992.

附录

程序编号	1	说明	距离相关系数计算	工具	Python

```
from scipy.io import loadmat, savemat

from scipy.spatial.distance import pdist, squareform
import numpy as np


def distcorr(X, Y):
    """距离相关系数"""
    X = np.atleast_1d(X)
    Y = np.atleast_1d(Y)
    if np.prod(X.shape) == len(X):
        X = X[:, None]
    if np.prod(Y.shape) == len(Y):
        Y = Y[:, None]
    X = np.atleast_2d(X)
    Y = np.atleast_2d(Y)
    n = X.shape[0]
    if Y.shape[0] != X.shape[0]:
        raise ValueError('Number of samples must match')
    a = squareform(pdist(X))
    b = squareform(pdist(Y))
    A = a - a.mean(axis=0)[None, :] - a.mean(axis=1)[:, None] + a.mean()
```

```
    B = b - b.mean(axis=0)[None, :] - b.mean(axis=1)[:, None] + b.mean()

    dcov2_xy = (A * B).sum() / float(n * n)
    dcov2_xx = (A * A).sum() / float(n * n)
    dcov2_yy = (B * B).sum() / float(n * n)
    dcor = np.sqrt(dcov2_xy) / (np.sqrt(np.sqrt(dcov2_xx) * np.sqrt(dcov2_yy))+0)
    return dcor

if __name__ == '__main__':
    fid = loadmat('activity_norm.mat')  # 加载归一化数据
    activity = fid['activity_norm']
    # activity1 = np.array(activity)
    fid = loadmat('descriptor_training_byvar.mat')
    descriptor_training_norm = fid['descriptor_training_byvar']

    #  计算相关系数
    coff = []
    coff1 = []
    for index, item in enumerate(descriptor_training_norm):
        coff.append(distcorr(activity[0], item))  # 与第一个活性的相关系数
        coff1.append(distcorr(activity[1], item))  # 与第二个活性的相关系数

    savemat('coff1.mat', {'coff1': coff1})
    savemat('coff.mat', {'coff': coff})
```

程序编号	2	说明	变量筛选	工具	Matlab

2.1

```
%% 问题1
[data_1_0,text_1]=xlsread('ERα_activity.xlsx');
[data_2_0,text_2]=xlsread('Molecular_Descriptor.xlsx');
%% 数据归一化
data_1=mapminmax(data_1_0',0,1)'; %mapminmax按行归一，需按列归一
%data_2_1=mapminmax(data_2_0',0,1)';
data_2_1=data_2_0;

%% 部分无用值剔除（即其值固定无波动）
Var_data_2=var(data_2_1); %计算方差
L_use=find(Var_data_2~=0); %定位
N=length(L_use);
data_2=data_2_1(:,Var_data_2~=0);
text_use=cell(N,1); %部分数据剔除后，对应分子名称更新
for k=1:N
```

```
    text_use{k,1}=text_2{1,L_use(k)+1};
end
L_unuse=find(Var_data_2==0);
N_unuse=length(L_unuse);
text_unuse=cell(N_unuse,1); %部分数据剔除后，对应分子名称更新
for k=1:N_unuse
    text_unuse{k,1}=text_2{1,L_unuse(k)+1};
end

%% 问题1——计算皮尔逊线性相关系数、斯皮尔曼秩（或Kendall）相关系数
N=length(data_2(1,:));
R_result_P=zeros(2,N); %记录线性相关系数
P_result_P=zeros(2,N); %记录线性相关显著性指标p值
R_result_S=zeros(2,N); %记录非线性相关系数
P_result_S=zeros(2,N); %记录非线性相关显著性指标p值
y1=data_1(:,1); %IC50_nM(值越小：代表生物活性越大，对抑制ERα活性越有效)
y2=data_1(:,2); %pIC50(值越大：表明生物活性越高,，对抑制ERα活性越有效)
for k=1:N
    x=data_2(:,k);
%     fig1=scatter(x,y1);
%     % saveas(fig1,strcat('1-',num2str(k),'.jpg'));
%     fig2=scatter(x,y2);
%     saveas(fig2,strcat('2-',num2str(k),'.jpg'));
    [R_result_P(1,k),P_result_P(1,k)]=corr(x,y1); % 计算IC50_nM的线性相关系数
    [R_result_P(2,k),P_result_P(2,k)]=corr(x,y2); % 计算IC50_nM的线性相关系数
    [R_result_S(1,k),P_result_S(1,k)]=corr(x,y1,'type','Spearman'); % 计算
IC50_nM的非线性相关系数
    [R_result_S(2,k),P_result_S(2,k)]=corr(x,y2,'type','Spearman'); % 计算
IC50_nM的非线性相关系数

end

%% 数据读取
[data_select_1,text_select]=xlsread('相关系数.xlsx','Sheet1');
[data_select_2,~]=xlsread('相关系数.xlsx','Sheet2');
[data_select_3,~]=xlsread('相关系数.xlsx','Sheet3');

%% 数据整合
data_select_use=[abs(data_select_1(:,1))';abs(data_select_2(:,1))';abs(data_s
elect_3(:,1))']; % 变成了3行
data_select_use_max=max(data_select_use); %取最大值  %%%%%%%%%%评判依据
data_select_use_max_sort=sort(data_select_use_max,2,'descend');
% 各类相关最大值排序
text_sort_2_M=cell(504,1);
```

```
data_sort_2_M=zeros(504,5);
k=1;
while k<=504
    L_x=data_select_use_max_sort(k);
    L_2=find(abs(data_select_use_max)==L_x);
    n=length(L_2);
    for g=1:n
        L_r=data_select_use_max(L_2(g));
        data_sort_2_M(k+g-1,1:2)=[L_r,L_2(g)];
        data_sort_2_M(k+g-1,3:5)=data_select_use(:,L_2(g))';
        text_sort_2_M{k+g-1,1}=text_select{L_2(g)+1,1};
    end
    k=k+n;
%
display(strcat(text{L_x,1},num2str(L_x),',',num2str(L_2),',',num2str(L_p)))
end
```

2.2

```
%% 计算分子间相关性、分类
corr_data_2=corr(data_2);
[M1,N1]=find(corr_data_2>=0.85 & corr_data_2<1 );
MN=[M1,N1];

%% 利用主成分分析
[coeff,score,latent] = pca(data_2); % 利用主成分分析
% 以贡献率为85%作为筛选的依据
w_sum=0;
n=0;
while w_sum<0.85
    n=n+1;
    w_sum=(w_sum*sum(latent)+latent(n))/sum(latent);
end
display(n) %最终n=29
```

2.3

```
%29类
KK=29;
data_data=cell(KK,1);
data_n=zeros(KK,1);
data_k=zeros(KK,1);
data_max=zeros(KK,1);
text_data=cell(KK,1);
for k=1:KK
    L=p{1,k};
```

```
    data_max(k)=max(data_select_use_max(L));
    data_use_2=data_2(:,L);
    Cor_data_use_2=corr(data_use_2,'type','Spearman');
    Cor_data_use_1=corr(data_use_2);
    text_data{k,1}=text_use(L,1);
    if min(min(Cor_data_use_2))<0.85 && min(min(Cor_data_use_1))<0.85
        display(k)
        data_k(k,1)=k;
        data_data{k,1}=data_use_2;
        [coeff,score,latent] = pca(data_use_2); % 利用主成分分析
        w_sum=0;
        n=0;
        while w_sum<0.85
            n=n+1;
            w_sum=(w_sum*sum(latent)+latent(n))/sum(latent);
        end
        display(n) %最终n=29
        data_n(k,1)=n;
        % save(strcat(num2str(k),'-',num2str(n),'.mat'),'data_use_2')
    end
end
```

2.4

```
% 29-X
load('for16_4.mat')
data_2_20=cell(29,1);
data_n_2_20=zeros(29,1);
data_k_2_20=zeros(29,1);
data_max_2_20=zeros(KK,1);
text_data_7=cell(6,1);
L_0=p{1,16};
for k=1:4
    L_1=p_2_20{1,k};
    L=L_0(L_1);
    data_max_2_20(k)=max(data_select_use_max(L));
    data_use_2=data_2(:,L);
    Cor_data_use_2=corr(data_use_2,'type','Spearman');
    Cor_data_use_1=corr(data_use_2);
    text_data_7{k,1}=text_use(L,1);
    if min(min(Cor_data_use_2))<0.85 && min(min(Cor_data_use_1))<0.85
        display(k)
        data_k_2_20(k,1)=k;
        data_2_20{k,1}=data_use_2;
        [coeff,score,latent] = pca(data_use_2); % 利用主成分分析
```

```
        w_sum=0;
        n=0;
        while w_sum<0.85
            n=n+1;
            w_sum=(w_sum*sum(latent)+latent(n))/sum(latent);
        end
        display(n) %最终n=29
        data_n_2_20(k,1)=n;
        save(strcat('19_2-',num2str(k),'-',num2str(n),'.mat'),'data_use_2')
    end
end
```

2.5

```
% x-x-x-x
load('for26-8.mat')
PP=p_2_20;
load('for26-8-8-2.mat')
data_2_20_8_6=cell(29,1);
data_n_2_20_8_6=zeros(29,1);
data_k_2_20_8_6=zeros(29,1);
data_max_2_20_8_6=zeros(KK,1);
text_data_7_6_5_2=cell(6,1);
L_0=p{1,26};
L_1=PP{1,8};
for k=1:2
    L_2=p_2_20{1,k};
    L_3=L_1(L_2);
    L=L_0(L_3);
    data_max_2_20_8_6(k)=max(data_select_use_max(L));
    data_use_2=data_2(:,L);
    Cor_data_use_2=corr(data_use_2,'type','Spearman');
    Cor_data_use_1=corr(data_use_2);
    text_data_7_6_5_2{k,1}=text_use(L,1);
    if min(min(Cor_data_use_2))<0.85 && min(min(Cor_data_use_1))<0.85
        display(k)
        data_k_2_20_8_6(k,1)=k;
        data_2_20_8_6{k,1}=data_use_2;
        [coeff,score,latent] = pca(data_use_2); % 利用主成分分析
        w_sum=0;
        n=0;
        while w_sum<0.85
            n=n+1;
            w_sum=(w_sum*sum(latent)+latent(n))/sum(latent);
        end
```

```
        display(n) %最终n=29
        data_n_2_20_8_6(k,1)=n;

save(strcat('19_2-1-2-',num2str(k),'-',num2str(n),'.mat'),'data_use_2')
    end
End
```

2.6

```
% x-x-x-x-x-x
load('for23_5.mat')
PP1=p_2_20;
load('for23-5-5-3.mat')
PP2=p_2_20;
load('for23-5-5-3-2-2.mat')
data_16_4_4_3_3_2=cell(29,1);
data_16_4_4_3_3_2_n=zeros(29,1);
data_16_4_4_3_3_2_k=zeros(29,1);
data_max_16_4_4_3_3_2=zeros(KK,1);
text_data_7_6_5_2_1_2=cell(6,1);
L_0=p{1,23};
L_1=PP1{1,5};
L_1_1=PP2{1,2};
for k=1:2
    L_2=p_2_20{1,k};
    L_3=L_1_1(L_2);
    L_4=L_1(L_3);
    L=L_0(L_4);
    data_max_16_4_4_3_3_2(k)=max(data_select_use_max(L));
    data_use_2=data_2(:,L);
    Cor_data_use_2=corr(data_use_2,'type','Spearman');
    Cor_data_use_1=corr(data_use_2);
    text_data_7_6_5_2_1_2{k,1}=text_use(L,1);
    if min(min(Cor_data_use_2))<0.85 && min(min(Cor_data_use_1))<0.85
        display(k)
        data_16_4_4_3_3_2_k(k,1)=k;
        data_16_4_4_3_3_2{k,1}=data_use_2;
        [coeff,score,latent] = pca(data_use_2); % 利用主成分分析
        w_sum=0;
        n=0;
        while w_sum<0.85
            n=n+1;
            w_sum=(w_sum*sum(latent)+latent(n))/sum(latent);
        end
        display(n) %最终n=29
```

```
        data_16_4_4_3_3_2_n(k,1)=n;

save(strcat('23-5-5-3-2-2-',num2str(k),'-',num2str(n),'.mat'),'data_use_2')
    end
End
```

2.7

```
% x-x-x-x-x-x
load('for2_20.mat')
PP1=p_2_20;
load('for2-20-7-4.mat')
PP2=p_2_20;
load('for2-20-7-4-3-4.mat')
PP3=p_2_20;
load('for2-20-7-4-3-4-1-3.mat')
data_16_4_4_3_3_2_x_x=cell(29,1);
data_16_4_4_3_3_2_x_x_n=zeros(29,1);
data_16_4_4_3_3_2_x_x_k=zeros(29,1);
data_max_16_4_4_3_3_2_x_x=zeros(KK,1);
text_data_7_6_5_2_1_2_1_2=cell(6,1);
L_0=p{1,2};
L_1=PP1{1,7};
L_1_1=PP2{1,3};
L_1_1_1=PP3{1,1};
for k=1:3
    L_2=p_2_20{1,k};
    L_3=L_1_1_1(L_2);
    L_4=L_1_1(L_3);
    L_5=L_1(L_4);
    L=L_0(L_5);
    data_max_16_4_4_3_3_2_x_x(k)=max(data_select_use_max(L));
    data_use_2=data_2(:,L);
    Cor_data_use_2=corr(data_use_2,'type','Spearman');
    Cor_data_use_1=corr(data_use_2);
    text_data_7_6_5_2_1_2_1_2{k,1}=text_use(L,1);
    if min(min(Cor_data_use_2))<0.85 && min(min(Cor_data_use_1))<0.85
        display(k)
        data_16_4_4_3_3_2_x_x_k(k,1)=k;
        data_16_4_4_3_3_2_x_x{k,1}=data_use_2;
        [coeff,score,latent] = pca(data_use_2); % 利用主成分分析
        w_sum=0;
        n=0;
        while w_sum<0.85
            n=n+1;
```

```
            w_sum=(w_sum*sum(latent)+latent(n))/sum(latent);
        end
        display(n) %最终n=29
        data_16_4_4_3_3_2_x_x_n(k,1)=n;

save(strcat('2-20-7-4-3-4-1-3-',num2str(k),'-',num2str(n),'.mat'),'data_use_2
')
    end
End
```

2.8

```
% x-x-x-x-x-x-x
load('for7_6.mat')
PP1=p_2_20;
load('for7-6-5-2.mat')
PP2=p_2_20;
load('for7-6-5-2-1-2.mat')
PP3=p_2_20;
load('for7-6-5-2-1-2-1-2.mat')
PP4=p_2_20;
load('for7-6-5-2-1-2-1-2-2-2.mat')
data_16_4_4_3_3_2_x_x=cell(29,1);
data_16_4_4_3_3_2_x_x_n=zeros(29,1);
data_16_4_4_3_3_2_x_x_k=zeros(29,1);
data_max_16_4_4_3_3_2_x_x=zeros(KK,1);
text_data_7_6_5_2_1_2_1_2_2_2=cell(6,1);
L_0=p{1,7};
L_1=PP1{1,5};
L_1_1=PP2{1,1};
L_1_1_1=PP3{1,1};
L_1_1_1_1=PP4{1,2};
L_zz=L_0(L_1(L_1_1(L_1_1_1(L_1_1_1_1))));
for k=1:2
    L_2=p_2_20{1,k};
    L=L_zz(L_2);
    data_max_16_4_4_3_3_2_x_x(k)=max(data_select_use_max(L));
    data_use_2=data_2(:,L);
    Cor_data_use_2=corr(data_use_2,'type','Spearman');
    Cor_data_use_1=corr(data_use_2);
    text_data_7_6_5_2_1_2_1_2_2_2{k,1}=text_use(L,1);
    if min(min(Cor_data_use_2))<0.85 && min(min(Cor_data_use_1))<0.85
        display(k)
        data_16_4_4_3_3_2_x_x_k(k,1)=k;
        data_16_4_4_3_3_2_x_x{k,1}=data_use_2;
```

```
        [coeff,score,latent] = pca(data_use_2); % 利用主成分分析
        w_sum=0;
        n=0;
        while w_sum<0.85
            n=n+1;
            w_sum=(w_sum*sum(latent)+latent(n))/sum(latent);
        end
        display(n) %最终n=29
        data_16_4_4_3_3_2_x_x_n(k,1)=n;

save(strcat('7-6-5-2-1-2-1-2-2-2-',num2str(k),'-',num2str(n),'.mat'),'data_use_2')
    end
End
```

2.9

```
% x-x-x-x-x-x-x-x
load('for7_6.mat')
PP1=p_2_20;
load('for7-6-5-2.mat')
PP2=p_2_20;
load('for7-6-5-2-1-2.mat')
PP3=p_2_20;
load('for7-6-5-2-1-2-1-2.mat')
PP4=p_2_20;
load('for7-6-5-2-1-2-1-2-2-2.mat')
PP5=p_2_20;
load('for7-6-5-2-1-2-1-2-2-2-2-2.mat')
data_16_4_4_3_3_2_x_x=cell(29,1);
data_16_4_4_3_3_2_x_x_n=zeros(29,1);
data_16_4_4_3_3_2_x_x_k=zeros(29,1);
data_max_16_4_4_3_3_2_x_x=zeros(KK,1);
L_0=p{1,7};
L_1=PP1{1,5};
L_1_1=PP2{1,1};
L_1_1_1=PP3{1,1};
L_1_1_1_1=PP4{1,2};
L_1_1_1_1_1=PP5{1,2};
L_zz=L_0(L_1(L_1_1(L_1_1_1(L_1_1_1_1(L_1_1_1_1_1)))));
text_data_7_6_5_2_1_2_1_2_2_2_2_2=cell(6,1);
for k=1:2
    L_2=p_2_20{1,k};
    L=L_zz(L_2);
    data_max_16_4_4_3_3_2_x_x(k)=max(data_select_use_max(L));
```

```
    data_use_2=data_2(:,L);
    Cor_data_use_2=corr(data_use_2,'type','Spearman');
    Cor_data_use_1=corr(data_use_2);
    text_data_7_6_5_2_1_2_1_2_2_2_2_2{k,1}=text_use(L,1);
    if min(min(Cor_data_use_2))<0.85 && min(min(Cor_data_use_1))<0.85
        display(k)
        data_16_4_4_3_3_2_x_x_k(k,1)=k;
        data_16_4_4_3_3_2_x_x{k,1}=data_use_2;
        [coeff,score,latent] = pca(data_use_2); % 利用主成分分析
        w_sum=0;
        n=0;
        while w_sum<0.85
            n=n+1;
            w_sum=(w_sum*sum(latent)+latent(n))/sum(latent);
        end
        display(n) %最终n=29
        data_16_4_4_3_3_2_x_x_n(k,1)=n;

save(strcat('7-6-5-2-1-2-1-2-2-2-1-2-',num2str(k),'-',num2str(n),'.mat'),'dat
a_use_2')
    end
end
```

程序编号	3	说明	AdaBoost预测模型数据处理	工具	Matlab

```
load('data_set_select');  %加载筛选出的20个变量
data=data_select_result;
activity = xlsread('ERα_activity.xlsx');
load('for-question2_and_question3.mat');  %加载测试数据
test=data_2_test_use_1;
test=mapminmax(test',0,1);  %测试数据归一化
test=test';
activity = activity(:,2);

[activity_norm,ps] = mapminmax(activity',0,1);  %数据归一化
activity_norm = activity_norm';

dim = 15;
[coeff,demension_dec,latent] = pca(data);  %PCA降维
data_dec = demension_dec(:,1:dim);  %降到dim维
[coff,test_demension_dec,latent_test] = pca(test);
data_dec = demension_dec(:,1:dim);  %降到dim维

test_data_dec = test_demension_dec(:,1:dim);
```

```
train_data = data_dec;
train_y = activity_norm;

test_data = test_data_dec;
save C:\Users\kkk\Desktop\数学建模\2021年D题\code\solution1\train_data;
save C:\Users\kkk\Desktop\数学建模\2021年D题\code\solution1\train_y;
save C:\Users\kkk\Desktop\数学建模\2021年D题\code\solution1\test_data;
save ps;  %保存归一化记录，用于反归一化

load('y_2.mat')
d=mapminmax('reverse',y_2,ps);  %反归一化
```

程序编号	4	说明	adaBoost模型建立	工具	python

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import GridSearchCV

import scipy.io as sio

#  加载数据
data_mat = sio.loadmat('train_data.mat')
reg_mat = sio.loadmat('train_y.mat')
X_train = data_mat['train_data']
y_train = reg_mat['train_y']

test_mat = sio.loadmat('test_data.mat')
X_test = test_mat['test_data']

# 用于训练和评估模型和调参
# X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1,
test_size=0.2)

######################决策树和adaboost回归器
regr_2 = AdaBoostRegressor(base_estimator=DecisionTreeRegressor(max_depth=11),
n_estimators=250, random_state=1, learning_rate=0.2, loss='linear')

#  网格搜索调参
# param_test1 = {'n_estimators': range(50, 1000, 50)}
# gsearch1 = GridSearchCV(estimator=regr_2, param_grid=param_test1)
# gsearch1.fit(X_train, y_train.ravel())
# print(f'{gsearch1.best_params_}+:{gsearch1.best_score_}')
```

```
#  训练回归模型
regr_2.fit(X_train, y_train.ravel())

# 预测
y_2 = regr_2.predict(X_test)
# print('adaBoost回归器:', regr_2.score(X_test, y_test.ravel()))

sio.savemat('y_2.mat', {'y_2': y_2})
```

程序编号	5	说明	用于CaCo-2分类预测	工具	Matlab

```
clc;clear;close all
load('data_set_select');
data=data_select_result;
load('for-question2_and_question3.mat');
test_data=data_2_test_use_1;
test_data=mapminmax(test_data',0,1);
test_data=test_data';
admet_training = xlsread('ADMET.xlsx');

dim = 5;
N_train=400;
admet_var = 1;  %用于ADMET第一列性质预测
[coeff,demension_dec,latent] = pca(data);
[coff,test_demension_dec,latent_test] = pca(test_data);
data_dec = demension_dec(:,1:dim);  %降到10维
test_data_dec = test_demension_dec(:,1:dim);
test=zeros(50,dim+1);
test(:,1:end-1) = test_data_dec;
test(:,end) =ones(50,1);

train =zeros(N_train,dim+1);
% test =zeros(1974-N_train,dim+1);

% data_dec = data(:,1:dim);
train(:,1:end-1)= data_dec(1:N_train,:);
train(:,end) = admet_training(1:N_train,admet_var);

% test(:,1:end-1) = data_dec(N_train+1:end,:);
save test;
save train;
% test(:,end) = admet_training(N_train+1:end,admet_var);

acc=KNNmain(80);
plot(N_train,acc,'*--','linewidth',1.5);
```

```
title('CaCo-2:KNN分类模型准确率随训练样本数变化的曲线');
xlabel('训练样本数')
ylabel('预测准确率')
legend('CaCo-2-KNN预测模型')
```

程序编号	6	说明	用于HOB分类预测	工具	Matlab

```
clc;clear;close all
load('data_set_select');
data=data_select_result;
load('for-question2_and_question3.mat');
test_data=data_2_test_use_1;
test_data=mapminmax(test_data',0,1);
test_data=test_data';
admet_training = xlsread('ADMET.xlsx');

dim = 8;
N_train=500;
admet_var = 4;  %只能用于ADMET4的预测
[coeff,demension_dec,latent] = pca(data);
[coff,test_demension_dec,latent_test] = pca(test_data);
data_dec = demension_dec(:,1:dim);  %降到dim维

test_data_dec = test_demension_dec(:,1:dim);
test=zeros(50,dim+1);
test(:,1:end-1) = test_data_dec;
test(:,end) =ones(50,1);



train =zeros(N_train,dim+1);
% test =zeros(1974-N_train,dim+1);

% data_dec = data(:,1:dim);
train(:,1:end-1)= data_dec(1:N_train,:);
train(:,end) = admet_training(1:N_train,admet_var);

% test(:,1:end-1) = data_dec(N_train+1:end,:);
save test;
save train;
% test(:,end) = admet_training(N_train+1:end,admet_var);

svm_data = data_dec;
svm_label = admet_training(:,admet_var);
save('./svm/svm_data.mat','svm_data','-mat');
save('./svm/svm_label.mat','svm_label','-mat');
```

```
acc=KNNmain(100);  %KNN分类
plot(N_train,acc,'*--','linewidth',1.5,'color','red');
title('HOB:KNN分类模型准确率随训练样本数变化的曲线');
xlabel('训练样本数')
ylabel('预测准确率')
legend('HOB-KNN预测模型')
```

程序编号	7	说明	KNNmain	工具	Matlab

```
function acc = KNNmain(k)
%%部分参考：
@https://github.com/jayshah19949596/Machine-Learning-Models/tree/master/K-Nearest%20Neigbour/Source%20Code
    %================================================
    %                 Loading Data
    %================================================
%     training_data = load(train_file);
%     testing_data = load(test_file);
    training_data = load('train.mat');
    testing_data = load('test.mat');
    train_target = training_data.train(:, end);
    test_target = testing_data.test(:, end);
    train_data = training_data.train(:, 1: end-1);
    test_data = testing_data.test(:, 1: end-1);
    %================================================
    %     Calculating mean and standard deviation
    %================================================
%     mean_of_dimension = mean(train_data);
%     std_deviation = std(train_data, 1);
%     %================================================
%     %                Normalising Data
%     %================================================
%     train_data = normalise(train_data, std_deviation, mean_of_dimension); % normalising training data
%     test_data = normalise(test_data, std_deviation, mean_of_dimension);   % normalising testing data
    %================================================
    %           Performing Knn calculation
    %================================================
    acc=knn(train_data, test_data, train_target, test_target, k);
end


function classification_accuracy = knn(train_data, test_data, train_target,
```

```
test_target, k)
    classification_accuracy = 0;
    for i = 1:size(test_data, 1)
        %=========================================
        %         Calculating Euclidean Distance
        %=========================================
        D = test_data(i, :)-train_data(: , :);
        D = D.^2;
        dist_mat = sum(D, 2);
        dist_mat = sqrt(dist_mat);
        dist = [dist_mat train_target];
        %=========================================
        %  Sorting Row according to minimum distance
        %=========================================
        dist = sortrows(dist, 1);
        %=========================================
        %       If K value is 1 Print Results
        %=========================================
        if k == 1
            k_neighbours = dist(k, :);
            predicted = k_neighbours(1, 2);
            true = test_target(i, 1);
            if true == predicted
                accuracy = 1;
                classification_accuracy = classification_accuracy + accuracy;
            else
                accuracy = 0;
            end
            fprintf('ID=%5d, predicted=%3d, true=%3d, accuracy=%4.2f \n', i,
predicted, true, accuracy)
        %=========================================
        % Else K value is greater then 1 Print Results
        %=========================================
        else
            k_neighbours = dist(1:k, :);
            if size(unique(k_neighbours(:, 2))) == 1
                predicted = unique(k_neighbours(:, 2));
            elseif size(unique(k_neighbours(:, 2))) == k
                predicted = k_neighbours(1, 2);
            else
                predicted = mode(k_neighbours(:, 2));
            end
            true = test_target(i, 1);
            if true == predicted
```

```
                accuracy = 1;
                classification_accuracy = classification_accuracy + accuracy;
            else
                accuracy = 0;
            end
            fprintf('ID=%5d, predicted=%3d, true=%3d, accuracy=%4.2f \n', i,
predicted, true, accuracy)
        end
    end
    classification_accuracy=classification_accuracy/size(test_target,1);
    fprintf('classification_accuracy=%6.4f \n', classification_accuracy)
end
```

程序编号	8	说明	用于CYP3A4、hERG、MN预测	工具	Matlab

```
clc;clear;close all
load('data_set_select');
data=data_select_result;
load('for-question2_and_question3.mat');
test_data=data_2_test_use_1;
test_data=mapminmax(test_data',0,1);
test_data=test_data';
admet_training = xlsread('ADMET.xlsx');

dim = 12;  %修改为对应降维数
admet_var = 2;  %用于ADMET第2，3，5列的预测，修改相应值
[coeff,demension_dec,latent] = pca(data);
data_dec = demension_dec(:,1:dim);  %降到dim维
[coff,test_demension_dec,latent_test] = pca(test_data);
test_data_dec = test_demension_dec(:,1:dim);
test_data=test_data_dec;

svm_data = data_dec;
length(svm_data)
svm_label = admet_training(:,admet_var);
train_data=svm_data;
train_label=svm_label;
save('./svm/svm_data.mat','svm_data','-mat');
save('./svm/svm_label.mat','svm_label','-mat');
save('./svm/train_data.mat','train_data','-mat');
save('./svm/train_label.mat','train_label','-mat');
save('./svm/test_data.mat','test_data','-mat');
```

程序编号	9	说明	SVM模型	工具	Python

```
from sklearn import svm
```

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import scipy.io as sio   # 用于加载mat文件
import pickle
from sklearn.model_selection import GridSearchCV

# 划分训练集和测试集
data_mat = sio.loadmat(r'svm_data.mat')
label_mat = sio.loadmat(r'svm_label.mat')

# train_data, test_data, train_label, test_label =
train_test_split(data_mat['svm_data'], label_mat['svm_label'],
#
random_state=1, test_size=0.2)

train_datamat = sio.loadmat('train_data.mat')
train_labelmat = sio.loadmat('train_label.mat')
test_datamat = sio.loadmat('test_data.mat')

train_data = train_datamat['train_data']
train_label = train_labelmat['train_label']
test_data = test_datamat['test_data']

# 训练svm分类器
classifier = svm.SVC(C=40, kernel='rbf', gamma=14,
decision_function_shape='ovr')
classifier.fit(train_data, train_label.ravel())

# param_test1 = {'kernel': ['rbf', 'linear', 'sigmoid', 'poly']}
# param_test1 = {'C': range(1, 100, 1)}
# param_test1 = {'gamma': range(1, 100, 1)}
# gsearch1 = GridSearchCV(estimator=classifier, param_grid=param_test1)
# gsearch1.fit(train_data, train_label.ravel())
# print(f'{gsearch1.best_params_}+:{gsearch1.best_score_}')

# 计算分类器的分类准确率
train_fit = classifier.predict(train_data)
test_fit = classifier.predict(test_data)
print(test_fit)
sio.savemat('model5_pridict.mat', {'test_fit': test_fit})
print("训练集：", accuracy_score(train_label, train_fit))
```

```
# print("测试集：", accuracy_score(test_label, test_fit))

model = pickle.dumps(classifier)  #保存模型
with open('svm.model', 'wb+') as f:
    f.write(model)
print("done")

# # 绘制混淆矩阵
# confusion_matrix = confusion_matrix(test_label, test_fit)
# N_class = 2
# classes = []
# for i in range(1, N_class+1):
#     classes.append(str(i))
# titles_options = [("Confusion matrix,without normalization", None),
#                   ("Normalized confusion matrix", 'true')]
#
# for title, normalize in titles_options:
#     disp = plot_confusion_matrix(classifier, test_data, test_label,
#                                  display_labels=classes,
#                                  cmap=plt.cm.Blues,
#                                  normalize=normalize)
#     disp.ax_.set_title(title)
# plt.show()
```

程序编号	10	说明	第四问优化模型	工具	Matlab

```
load('data_2.mat')
[data_3_0,text_3]=xlsread('ADMET.xlsx'); %ADMET性质
[data_1_0,text_1]=xlsread('ERα_activity.xlsx'); %活性
data_3_1=zeros(size(data_3_0));
data_3_1(:,[1,2,4])=data_3_0(:,[1,2,4]);
data_3_1(:,[3,5])=abs(data_3_0(:,[3,5])-1);
z_sum=sum(data_3_1,2);
L_use_select=[440,55,11,353,102,4,5,366,419,493,174,91,56,482,284,439,99,42,2
21,406];
y2=data_1_0(:,2);
data_select_result=data_2(:,L_use_select);
F1=quantile(y2,0.5,1); %抑制ERα是否具有更好的生物活性的临界指标
L_1=find(y2>=F1 & z_sum>=3);
L_2=find(y2<F1 | z_sum<3);
data_CC1=data_select_result(L_1,:);
data_CC2=data_select_result(L_2,:);
H=zeros(4,20);
for k=1:20
    x1=data_CC1(:,k);
```

```
    x2=data_CC2(:,k);
    H(1,k) = jbtest(x1,0.05);
    H(2,k) = jbtest(x2,0.05);
    H(3,k) = kstest2(x1,x2);
    H(4,k) = ttest2(x1,x2);
    fig2=figure;
    subplot(2,2,1);hist(x1,30);title('x1')
    subplot(2,2,2);hist(x2,30);title('x2')
    subplot(2,2,3);plot(x1,'*');title('x1')
    subplot(2,2,4);plot(x2,'*');title('x2')
    saveas(fig2,strcat('20-',num2str(k),'.jpg'));
end

%% 临界值查找
k=1;
R_result_P_x=zeros(4,100);
P_1=0.5;
P_2=0.5;
x1=data_CC1(:,k);
x2=data_CC2(:,k);
Z_1=quantile(x1,P_1,1);
Z_2=quantile(x2,P_2,1);
R_result_P_x(:,1)=[P_1;P_2;Z_1;Z_2];
if Z_1>Z_2
    K_z=1;
else
    K_z=-1;
end
n=0;
while K_z*(Z_1-Z_2)>0
    n=n+1;
    P_1=P_1-K_z*0.01;
    P_2=P_2+K_z*0.01;
    Z_1=quantile(x1,P_1,1);
    Z_2=quantile(x2,P_2,1);
    R_result_P_x(:,n+1)=[P_1;P_2;Z_1;Z_2];
end
display(strcat('迭代',num2str(n),'次'))
```