

基于原子特性知识增强的分子毒性预测方法

方舒言¹, 刘宇^{1,2*}, 侯阿龙¹, 秦欢欢³, 刘嵩^{3,4}

- (1. 武汉科技大学 计算机科学与技术学院, 湖北 武汉 430072;
2. 湖北省智能信息处理与实时工业系统重点实验室, 湖北 武汉 430072;
3. 武汉科技大学 医学院, 湖北 武汉 430072;
4. 湖北省职业危害识别与控制湖北省重点实验室, 湖北 武汉 430072)

摘要:当前基于深度学习的化学分子毒性预测方法主要利用了分子的字符串表示,但现有的字符串表示模型忽视了分子中不同原子的特性知识,从而导致学习模型未能充分利用领域知识。针对上述问题,提出了显式引入氢原子及利用摩根指纹半径增强原子特性知识的方法,使得毒性预测模型能够学习到化学分子中原子的特性知识。在改进的毒性预测模型中,用氢原子及原子特性知识增强的分子摩根指纹标识符序列作为输入,并在嵌入层额外引入了分子摩根指纹的半径特征。为了验证方法的有效性,对预训练后的模型在主流的毒性预测数据集 Tox21 上进行了微调 and 测试。实验结果表明,相比于现有的基于分子序列的化学分子毒性预测方法,改进的方法在多个通道上取得了最佳的 AUC 分数。

关键词:分子毒性预测;自监督学习;知识增强;药物发现;摩根指纹

中图分类号: TP183

文献标识码: A

文章编号: 1673-629X(2024)03-0155-08

doi: 10.3969/j.issn.1673-629X.2024.03.023

A Molecular Toxicity Prediction Method Based on Knowledge Enhancement of Atomic Properties

FANG Shu-yan¹, LIU Yu^{1,2*}, HOU A-long¹, QIN Huan-huan³, LIU Song^{3,4}

- (1. School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430072, China;
2. Hubei Key Laboratory of Intelligent Information Processing and Real-time Industrial Systems, Wuhan 430072, China;
3. School of Medicine, Wuhan University of Science and Technology, Wuhan 430072, China;
4. Hubei Key Laboratory of Occupational Hazard Identification and Control in Hubei Province, Wuhan 430072, China)

Abstract: Current deep learning-based methods for toxicity prediction of chemical molecules mainly utilize the string representation of molecules, but the existing string representation models ignore the knowledge of the properties of different atoms in molecules, which leads to the failure of learning models fully utilizing the domain knowledge. To address these problems, a method that explicitly introduces hydrogen atoms and enhances the knowledge of atomic properties using the Morgan fingerprint radius is proposed to enable the toxicity prediction model to learn the knowledge of the properties of atoms in chemical molecules. In the improved toxicity prediction model, a sequence of molecular Morgan fingerprint identifiers enhanced with hydrogen atoms and atomic property knowledge is used as input, and the radius feature of molecular Morgan fingerprint is additionally introduced in the embedding layer. To validate the effectiveness of the proposed method, the pre-trained model was fine-tuned and tested on the mainstream toxicity prediction dataset Tox21. The experimental results showed that the improved method achieved the best AUC scores on multiple channels compared with the existing molecular sequence-based chemical molecule toxicity prediction methods.

Key words: molecular toxicity prediction; self-supervised learning; knowledge enhancement; drug discovery; Morgan fingerprint

0 引言

新药研发存在周期长、费用高和成功率低等特点。人工智能技术是近些年的热点技术之一,在很多领

域都有非常广泛的应用,多种人工智能方法已经成功应用于药物的发现过程^[1]。当前化学分子毒性预测方法主要从分子结构的序列表示中学习特征,如

收稿日期: 2023-01-26

修回日期: 2023-05-30

基金项目: 国家自然科学基金资助项目(U1836118, 62261023); 湖北省职业危害识别与控制湖北省重点实验室开放项目(OHIC2019G06)

作者简介: 方舒言(1997-), 男, 硕士研究生, 研究方向为人工智能; 通信作者: 刘宇(1980-), 男, 副教授, 博士研究生, CCF 会员(16111M), 研究方向为知识工程、智能系统。

SMILES^[2] (Simplified Molecular Input Line Entry System)。SMILES 通过对分子图的生成树实施深度优先的前序遍历,为每个原子、键、树遍历决策和断环生成符号,定义了分子的字符串表示,所得到的字符串对应于分子图生成树的序列化。因比其他表示结构的方法(包括图)更紧凑,SMILES 已被广泛用于分子毒性预测。

目前,以 Chemberta^[3] 为代表的分子毒性预测方法借鉴了自然语言处理技术的思想,直接使用大量无标注的 SMILES 作为语料库来学习分子表征,并使用学习到的分子表示用于毒性预测。然而 SMILES 语法复杂且限制性强,常规字符集上的大多数序列不能明确定义分子。

由于上述限制,基于 SMILES 的深度学习模型只能将学习重点放在分子串的语法上,导致现有的毒性预测方法仅损失了分子中的原子特性等知识。针对上述问题,以 Mol-BERT^[4] 为代表的方法选择了使用摩根指纹标识符(Morgan Fingerprint Identifier, MFI)序列作为输入来进行模型的训练,并在 MFI 融入了原子特性的知识(如原子序数、电荷量、原子的度等)。摩根指纹(Morgan Fingerprints)是一种使用哈希算法生成的圆形分子指纹^[5-6]。其中,原子的 MFI 是对原子及其属性(如原子序数、电荷量等)进行哈希得到的一个定长整数,而子结构 MFI 则是对分子中某原子 MFI 及以该原子为中心一定半径范围内的其他原子的 MFI 进行联合哈希得到的定长整数。根据设置的指纹半径的不同,可以得到蕴含不同知识的 MFI,比如以某原子为中心且半径为 0 生成的 MFI 仅包含该原子的知识,以某原子为中心且半径为 1 生成的 MFI,则蕴含该原子及相邻原子子结构知识。

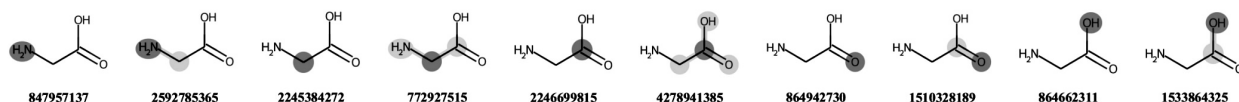
根据上述对摩根指纹及 MFI 的说明可知,Mol-

BERT 虽在一定程度上融入了原子特性的知识,但是仍然存在局限。比如,Mol-BERT 的输入 MFI 序列中未显式引入氢原子的知识,而且对于 MFI 序列的半径信息 Mol-BERT 方法并没有进行显式的编码。

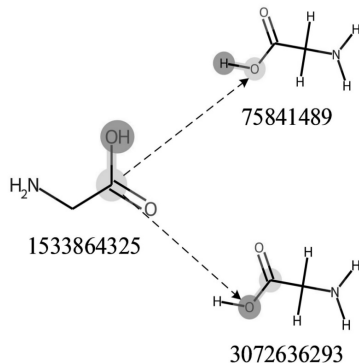
针对目前基于分子序列的毒性预测方法均未考虑氢原子和摩根指纹半径等领域知识的问题,该文提出了一种基于原子特性知识增强的分子毒性预测方法。一方面,改进的分子毒性预测方法考虑了分子中所有氢原子,输入的 MFI 序列显式引入了氢原子的 MFI 和以氢原子为中心的子结构的 MFI。在考虑了分子中所有的氢原子之后,子结构 MFI 所代表的子结构粒度更小,模型的 MFI 字典大小变为引入氢原子之前的约一半。这是由于在同样的半径限制下,使得原本一些大而稀有的子结构因为氢原子的引入将被转换成小而常见的子结构,如图 1(b)所示。该现象类似于自然语言处理领域的 WordPiece^[7],它将单词拆成词根和词缀,一些生僻词被拆成常见词根和词缀的组合,能够减小自然语言处理模型的词表大小。另一方面,受 SongNet^[8] 的启发,为了让模型学习 MFI 序列时可以区分不同半径的 MFI,提出使用摩根指纹半径的知识扩展嵌入层的方法,使模型学习到的原子 MFI 和子结构 MFI 有所区分。图 1(a)~(c)描述了氢原子和摩根指纹半径知识增强 MFI 序列的过程及增强后 MFI 序列与原本 MFI 序列的区别。该文的主要贡献包括 3 个方面:

(1) 提出生成分子 MFI 序列时显式地引入氢原子的方法,使得分子毒性预测模型可以学习到氢原子信息,同时减少了 MFI 字典的大小和模型的参数量;

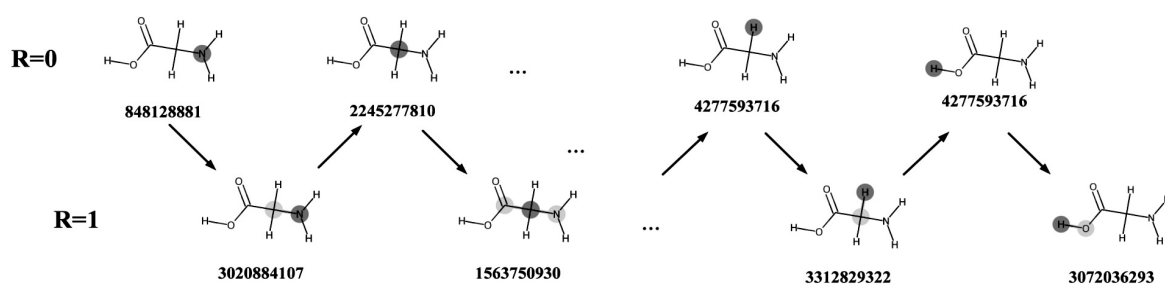
(2) 改进的分子毒性预测方法,额外考虑了各原子及相邻原子构成的子结构的知识,并在模型的嵌入层强化了摩根指纹半径知识;



(a) 原本的 MFI 序列



(b) 显式引入氢原子之后相应的原子邻域粒度细化



(c) 使用氢原子和摩根指纹半径知识增强的 MFI 序列

图1 氢原子和摩根指纹半径知识增强的 MFI 序列

(3) 为了验证改进方法的有效性,利用 Pubchem^[9] 分子库中的 1 000 万分子预训练了基于原子特性增强的分子毒性预测模型 (Hydrogen and Radius enhanced Toxicity Prediction model, HRToxPred), 并在主流的毒性预测数据集 Tox21^[10] 上进行了验证。

1 相关工作

在基于深度学习的分子毒性预测领域中,主流方法直接在分子的 SMILES 字符串上训练模型来预测分子毒性。表 1 总结了基于分子序列的毒性预测模型的相关工作。FP2VEC^[11] 和 Mol2vec^[12] 使用了自然语言处理领域的 Word2vec^[13] 思想。其中, Mol2vec 学习的是 MFI 的低维向量表示,然后输入学习到的 MFI 向量表示至下游网络中预测分子毒性,而 FP2VEC 学习的是比特分子指纹比特串的低维向量表示。然而,在自然语言处理领域中 Word2vec 的词和向量是一对一的关系,所以无法解决多义词的问题。在化学领域,同样

存在类似“一词多义”的场景,例如相同的原子所处的环境不同其作用也不同,所以 FP2VEC, Mol2vec 存在着类似的问题。Chemberta 和 Mol-BERT 使用了基于 Transformer 的掩码语言模型。其中, Chemberta 设计了一种 SMILES 的分词方法,使用 Roberta^[14] 的预训练策略训练了基于 SMILES 的分子毒性预测模型,然而 SMILES 的分词方法需要化学领域的专家来设计,需要消耗大量人力。Mol-BERT 使用分子的 MFI 序列作为语料库,使用 BERT^[15] 的 MLM (Masked Language Model) 策略预训练了基于 MFI 序列的分子毒性预测模型。相比于 Chemberta, Mol-BERT 所使用的 MFI 序列不需要人工分词。然而 Mol-BERT 使用的 MFI 序列未显式引入氢原子且序列中原子指纹与子结构指纹混杂在一起,使模型学习到的分子表示并未区分原子指纹与子结构指纹。针对上述方法存在的问题,该文提出生成 MFI 序列时显式引入氢原子并对半径信息进行显式编码的知识增强方法。

表1 相关工作总结

序号	模型类别	模型名称	分子数据类型	优点	缺点
1	Word2vec	FP2VEC	分子指纹	(1) 通用性强;	(1) 特定的任务无法优化;
2		Mol2vec	隐式氢摩根指纹	(2) 考虑了序列上下文	(2) 无法解决“一词多义”的场景
3	Transformer	Chemberta	SMILES	(1) 免去了数据处理; (2) 可以处理“一词多义”	(1) SMILES 分词困难; (2) 学习重点在于分子字符串的语法
4		Mol-BERT	隐式氢摩根指纹	(1) 考虑了原子或子结构的环境; (2) 可以处理“一词多义”	(1) MFI 序列半径特征未显式编码; (2) 未显式考虑分子中氢原子

2 方法

2.1 摩根指纹标识符序列

为了便于解释显式引入氢原子后 MFI 的计算方法,现给出以下定义:

定义1 原子邻域 (Atom Neighborhood): 给定一个分子图 $mg = (A, B)$, A 为该分子中的所有原子的集合, B 为该分子中所有的化学键的集合。对于一个原子 $a \in A$, 给定半径 r , 以 a 为中心, r 为半径的原子邻域定义为 $nbr(mg, a, r) = \{[a, a_i, b] \mid a \in A, a_i \in A, b \in B, a \neq a_i, \text{dist}(a, a_i) \leq r, b = \langle a, a_i \rangle\}$, 其中 a_i 为

与 a 通过化学键 b 相连的原子, $\text{dist}(\cdot)$ 表示分子图中两个原子之间的跳数。

将原子邻域的半径设为 1, 计算 MFI 时需考虑到两种不同规模的原子邻域 $nbr(mg, a, 0)$ 和 $nbr(mg, a, 1)$ 。如图 2 所示, 深色节点对应 $nbr(mg, a, 0)$, 而由深色节点 a 和浅色节点 a_i 组成的子结构对应 $nbr(mg, a, 1)$ 。由此, 原子的 MFI 和子结构的 MFI 将由其对应的原子邻域特性经过哈希得到。

定义2 摩根指纹^[5-6] (Morgan Fingerprint): 摩根指纹是一个稀疏的比特串。一个分子可以通过散列分子中每个原子的邻域信息得到一个索引, 该索引位置

的比特位被置为 1。

定义 3 摩根指纹标识符 (Morgan Fingerprint Identifier, MFI): 摩根指纹标识符是一个固定长度的整数, 用来表示原子邻域的特征。通常情况下, 一个原子邻域的摩根指纹标识符可以对该原子邻域的属性 (原子序数、电荷量等) 进行哈希来获得。记为 $mfi_{nbr(mg, a, r)} = Hash(Attrs(nbr(mg, a, r)))$, $Hash(\cdot)$ 为哈希函数, $Attrs(\cdot)$ 表示获取原子邻域的属性。

原子邻域、摩根指纹和摩根指纹标识符之间的关系如图 2 所示。对于一个分子经过算法 1 的计算, 可以获得该分子的 MFI 字典。字典的键为 MFI, 字典的值为由该 MFI 对应原子邻域中心原子的原子索引和原子邻域半径组成的元组 $[a, Index, r]$ 。该分子的摩根指纹则是一个长的比特串, 其中索引为 MFI 字典中键值的比特位被置为 1, 如图 2 箭头指示。

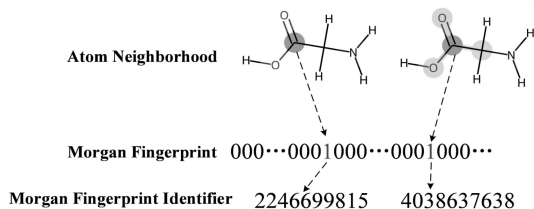


图 2 原子邻域、摩根指纹、摩根指纹标识符之间的关系

定义 4 原子的摩根指纹标识符: 给定一个分子 $mg = (A, B)$ 和一个原子 $a \in A$, 该原子的摩根指纹标识符可以通过哈希原子邻域 $nbr(mg, a, 0)$ 的属性得到。

算法 1

输入: 分子图 $mg = (A, B)$, 半径 r

输出: 存储该分子 MFI 的字典 D

Initialization;

/* 初始化一个容器存所有的 mfi, 偏移量为原子索引 */

Vector<Integer>invars;

for $a \in A$ do

/* 初始化一个容器存 Attrs($nbr(mg, a, 0)$) */

Vector<Integer>components;

components.pushback(a .AtomicNum); // 原子序数

components.pushback(a .TotalDegree); // 原子的度

if $a \neq \text{Hydrogen}$ then

// 非氢原子连接的氢原子数

components.pushback(a .TotalNumHs);

end

components.pushback(a .FormalCharge); // 原子的电

荷量

invars.pushback(boost:hash(components));

end

/* 将 $mfi_{nbr(mg, a, 0)}$ 的信息存入字典 */

for $a \in A$ do

$D[invars[a, Index]].pushback(Tuple(a, Index,$

$0))$;

end

/* 初始化容器存 $mfi_{nbr(mg, a, 1)}$ */

Vector<Tuple>nbrs;

for $a \in A$ do

/* 获取从 a 出发的化学键 */

bonds = getChemicalBonds(a);

for $\langle a, a_i \rangle \in \text{bonds}$ do

if $a \neq \text{Hydrogen}$ then

nbrs.pushback(Tuple($\langle a, a_i \rangle$.type

invars[a_i .Index]));

end

end

invar = 0;

boost:hashcombine(invar, invars[a .Index]);

for $nbr \in \text{nbrs}$ do

boost:hashcombine(invar, nbr);

end

/* 将 $mfi_{nbr(mg, a, 1)}$ 的信息存入字典 */

$D[invar].pushback(Tuple(a, Index, 1))$;

end

该文考虑的原子邻域 $nbr(mg, a, 0)$ 的属性包含原子序数、电荷量、非氢原子连接的氢原子数、原子在分子图中的度。算法 1 的第 3 ~ 18 行具体描述了该计算摩根原子 MFI 的过程。

定义 5 子结构的摩根指纹标识符: 给定一个分子 $mg = (A, B)$ 和一个原子 $a \in A$, 以 a 为中心的子结构的摩根指纹标识符可以通过哈希原子邻域 $nbr(mg, a, 1)$ 的属性得到。

该文考虑的原子邻域 $nbr(mg, a, 1)$ 的属性包含该原子邻域中化学键 b 的类型、该原子邻域中原子 a 和 a_i 的摩根指纹。算法 1 的 19 ~ 37 行具体描述了计算过程。

定义 6 摩根指纹标识符序列: 摩根指纹标识符序列是由一个分子中所有的原子邻域的摩根指纹标识符组成的, 摩根指纹标识符的顺序按照 SMILES 定义的原子顺序排列。如果对于分子中某原子 a 有多个摩根指纹标识符, 则按照 r 由小到大进行排序。

根据以上定义, 算法 1 描述了计算最大半径设置为 1 时的分子的 MFI 的过程。

2.2 数据预处理

数据预处理阶段, 因为同一个分子可以有多种不同的 SMILES 形式, 所以首先要对分子 SMILES 进行标准化。SMILES 之所以出现不同是因为可以随意修改原子的“出场顺序”以得到不同的 SMILES。该文使用 Rdkit^[16-17] 实现了标准化过程, 并根据分子化合价态平衡原理补全 SMILES 中的氢原子。最后使用算法 1 计算并排序生产分子的 MFI 序列。

对于一个分子, 其原子个数为 N , 使用算法 1 生成

最大半径为 R 的原子邻域的 MFI, 用 $\text{mfi}_i^r (0 \leq i \leq N, 0 \leq r \leq R)$ 表示, 其中下标 i 表示原子的索引, 上标 r 表示当前指纹标识符代表的子结构半径。如图 2 所示, mfi_i^0 (即深色节点) 表示按原子顺序排序的第 i 个原子的 MFI, 而 mfi_i^1 (深色节点与浅色节点组成的子结构) 表示当前原子与其相邻的原子构成的子结构的 MFI。以甘氨酸(NCC(=O)O)为例, 使用算法 1 计算该分子的 MFI 并排序生成了最大半径为 1 的甘氨酸分子 MFI 序列。如图 1(c) 所示, 箭头指示的序列为模型输入的序列。整个分子的序列可表示为 $\text{mfi}_0^0 \text{mfi}_0^1 \text{mfi}_1^0 \text{mfi}_1^1 \cdots \text{mfi}_{19}^0 \text{mfi}_{19}^1 \text{mfi}_{20}^0 \text{mfi}_{20}^1$ 。

根据上述方法, 该文选择 Pubchem 分子库中 1 000 万分子的 SMILES, 并使用算法 1 分子的 MFI 并排序形成 MFI 序列。整个过程中有关分子的处理都使用 Rdkit 分子处理工具实现。统计 1 000 万分子的是否显式引入氢原子的 MFI 序列, 得到不同 MFI 序列的 MFI 字典信息, 如表 2 所示。

表 2 是否引入氢的 MFI 字典大小比较

MFI 字典	字典大小
未显式引入氢	92 461
显式引入氢	41 854

显式引入氢原子 MFI 之后, 分子中的某些相对较大的子结构标识符可以分解为一些相对较小的子结构的组合, 例如图 1 中以该羟基(-OH)上氧原子为中心

半径为 1 的子结构会被拆分成以羟基中氢原子为中心半径为 1 的羟基结构和以氧原子为中心半径为 1 的结构。这类似于自然语言处理领域把单词拆成词根与词缀的方法。因此, 引入显式表示氢原子之后, 生成的 MFI 字典大小更小且 MFI 对应的原子邻域粒度更细。该文尝试扩展算法 1 的最大半径限制, 当最大半径达到 2 时, 即便引入了氢原子一定程度上减小了 MFI 字典的规模, 然而最大半径限制为 2 时 MFI 字典的大小超过了 100 万, 因此该文仅考虑最大半径限制为 1 的情况。

2.3 模型架构

HRToxPred 包含 Embedding 模块、Transformer 模块和下游任务模块, 如图 3 所示。图中 M 为输入的 MFI, F 为 MFI 嵌入(Token 嵌入), P 为位置编码(Position 嵌入), R 为半径嵌入(Radius 嵌入)。因为生成的每个 MFI 序列代表一个分子, 而每个分子都是单独的个体, 无需考虑两个分子之间的关系, 所以文中的预训练去掉了 BERT 模型预训练策略中“下一句预测”(Next Sentence Prediction, NSP) 预训练任务, 输入样本变为单个 MFI 序列, 因此 BERT 模型原本用来区分两个句子的 Segment 编码失去了意义。该文修改了 BERT 的 Embedding 模块, 将 Segment 嵌入改为摩根指纹半径嵌入来区分 MFI 的类别, 最终 Embedding 模块包含半径嵌入、位置嵌入和 MFI 编码嵌入。

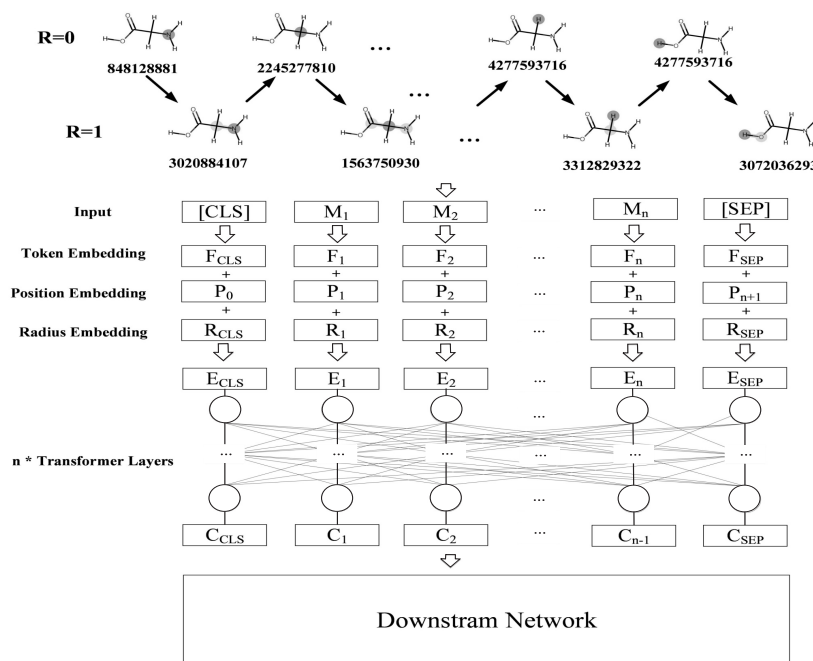


图 3 模型架构示意图

这里以图 1 中的分子为例来解释 HRToxPred 方法细节。对于输入数据, 首先进行甘氨酸 SMILES 的标准化, 然后使用 Rdkit 向甘氨酸分子对象中补充氢原子, 最后使用算法 1 的方法计算生成分子的 MFI 并

排序得到 MFI 序列, 并在序列中加入 [CLS] 等特殊编码。最后生成的序列为 $[\text{CLS}] \text{mfi}_0^0 \text{mfi}_0^1 \text{mfi}_1^0 \text{mfi}_1^1 \cdots \text{mfi}_{19}^0 \text{mfi}_{19}^1 \text{mfi}_{20}^0 \text{mfi}_{20}^1 [\text{SEP}]$ 。在计算生成某个 MFI 的同时可以获得该 MFI 对应的半径, 通过此信息进行半

径编码获得半径特征序列 $R_{\text{CLS}} R_1 R_2 \cdots R_{39} R_{40} R_{\text{SEP}}$, 其中 R_i 为 MFI 序列中索引为 i 的 MFI 对应的半径编码, $R_{\text{CLS}}, R_{\text{SEP}}$ 为摩根指纹半径编码之外的特殊编码。最后根据每个标识符在句中的顺序生成每个 MFI 的绝对位置编码 $P_0 P_1 P_2 \cdots P_{43}$ 。

经过上述步骤输入数据就处理完成, 接下来进行嵌入。对于输入特征, 首先要获得 MFI 编码、半径编码、位置编码的嵌入, 通过把所有嵌入向量相加的方式, 公式如下:

$$H_t = E_{mt} + E_{pt} + E_{rt} \quad (1)$$

式中, t 为序列中 MFI 的索引, E_{mt} , E_{pt} , E_{rt} 分别为 MFI 嵌入、位置嵌入和半径嵌入, 即通过半径嵌入 E_{rt} 区分了原子 MFI 和子结构 MFI。处理完输入数据和嵌入之后, HRToxPred 使用自注意力机制进行学习。

$$q_i = W^Q H_i \quad (2)$$

$$k_i = W^K H_i \quad (3)$$

$$v_i = W^V H_i \quad (4)$$

$$c_i = \text{self-att}(q_i, k_i, v_i) \quad (5)$$

式中, W^Q , W^K , W^V 为参数矩阵, $\text{self-att}(\cdot)$ 为注意力函数, C_i 为该注意力头输出。根据上述方法将得到的所有注意力头的 C_i 向量进行拼接, 然后与参数矩阵 W 求积得到输出 c , 再将 c 输入到激活函数中得到对应的输出 o 。在下游任务中只需将 o 输入到下游任务的分类器网络中即可。式 1~7 描述了模型的计算过程。

$$c = \text{Concat}(c_i) W \quad (6)$$

$$o = \max(0, ZW_1 + b_1) W_2 + b_2 \quad (7)$$

3 实验与分析

3.1 基准模型

选取了当前被广泛使用的基于分子序列的毒性预测模型 Chemberta 和 Mol-BERT 作为对比模型。其中, Chemberta 基于 Roberta 模型实现。预训练时, 使用 1 000 万分子的 SMILES 作为语料库, 随机遮蔽了每个输入字符串中 15% 的 Token。在学习恢复 Mask Token 时, 该模型形成了一个化学空间的分子结构表示。Mol-BERT 使用未显式引入氢原子 MFI 和摩根指纹半径特征的 MFI 序列作为输入。Mol-BERT 同样使用了 1 000 万分子的 MFI 序列, 并利用了 BERT 的架构及预训练策略进行了预训练, 最后将预训练好的模型进行下游的毒性预测任务。

3.2 预训练

HRToxPred 与主流的自然语言处理预训练语言模型的方式不同, 去掉了 NSP 预训练任务, 仅留 MLM 任务。具体来说, 该文使用 bert-base 的模型设置, 并将分子 MFI 序列中随机 15% 的标识符屏蔽为 [MASK]。

此外, 预训练使用了动态掩码, 即同一个 MFI 序列会产生不同的掩码序列, 这样可以重复使用预训练样本, 而且可以防止过拟合。

3.3 微调

选择主流毒性预测数据集 Tox21 的 12 个分类任务进行微调。数据集详情见表 3。

表 3 Tox21 各任务细节

任务	分子总数	正样本数	负样本数
sr_p53	6 774	423	6 351
sr_are	5 832	942	4 890
sr_atad5	7 072	264	6 808
sr_hse	6 467	372	6 095
sr_mmp	5 810	918	4 892
nr_ahr	6 549	768	5 781
nr_er	6 193	793	5 400
nr_ar	7 265	309	6 956
nr_ar_lbd	6 758	237	6 521
nr_er_lbd	6 955	350	6 605
nr_ppar_gamma	6 450	186	6 264
nr_aromatase	5 821	300	5 521

3.4 评估标准

由于该文选取的数据集正负样本不平衡, 而精度、准确率、召回率等指标受正负样本不平衡的影响, 不能客观反映出模型的性能, 且在分子毒性预测中区分正负样本的概率阈值并不确定, 所以选择 AUC^[18] (Area Under Curve) 作为评测指标可避免以上因素带来的影响。

AUC 被定义为 ROC (Receiver Operating Characteristic) 曲线下的面积。ROC 曲线, 又称接受者操作特征曲线。二分类任务分类阈值的设定不同会得出不同的真阳率 (True Positive Rate, TPR) 和假阳率 (False Positive Rate, FPR), 将同一模型每个阈值的 (FPR, TPR) 坐标都画在 ROC 空间里, 就成为特定模型的 ROC 曲线。AUC 计算公式如下:

$$\text{AUC} = \frac{\sum (p_i, n_j)_{p_i > n_j}}{P * N} \quad (8)$$

式中, P 为正样本数量, N 为负样本数量, p_i 为正样本预测得分, n_j 为负样本预测得分。

3.5 实验设置

HRToxPred 包含 Embedding 模块、Transformer 模块和下游任务模块, 模型的实现框架使用 UER-py^[19]。

Embedding 模块包含 Token 嵌入、Position 嵌入和半径嵌入。HRToxPred 设置的输入序列长度与 BERT 保持一致均为 512 个 Token。

Transformer 模块选择 bert-base 结构, bert-base 包

含 12 个 Transformer 层和 12 个自注意力头,隐藏层维度为 768。

下游任务模块的微调网络使用 UER-py 框架中默认的网络,即在预训练完成的模型后接 2 层全连接层。对于每个数据集,按照 80/10/10 的比例随机划分为训练集/验证集/测试集,并进行了 32 个 epoch 的微调,具体参数见表 4。

表 4 模型参数设置

参数	数值
学习率	2e-5 ~ 5e-5
批次大小	8-16
训练轮数	32
隐藏层维度	768
注意力头	12
分类器全连接层数	2

3.6 消融实验分析

为了进一步验证 HRToxPred 中引入的摩根指纹半径特征及考虑氢原子的作用,进行了消融实验,即用 Pubchem 中的 1 000 万分子预训练如下四个模型:

- Base: 输入 MFI 序列不显式引入氢原子及其子结构 MFI,嵌入层没有额外的摩根指纹半径嵌入;
- AddHS: 输入 MFI 序列不显式引入氢原子及其子结构 MFI,嵌入层没有额外的摩根指纹半径嵌入;
- AddRadius: 输入 MFI 序列不显式引入氢原子及其子结构 MFI,嵌入层有额外的摩根指纹半径嵌入;
- HRToxPred: MFI 序列显式引入氢原子及其子结构 MFI,嵌入层有额外的摩根指纹半径嵌入。

然后将上述四个模型在 Tox21 上微调。如图 4 所示,使用显式引入氢原子 MFI 的模型预训练收敛得更快,主要原因是 HRToxPred 考虑了氢原子及其子结构

MFI 造成的词表减小。另外,从表 5 中 Tox21 不同通道上的 AUC 分数对比来看,仅使用考虑了氢原子 MFI 序列作为输入(AddHS)和仅引入摩根指纹半径嵌入(AddRadius)都可以提升模型的分类能力,而两者的组合对模型分类能力的提升更显著。HRToxPred 相比于 Base 模型,在 sr_p53, nr_ahr, nr_er, nr_ar, sr_are, nr_ar_lbd, nr_er_lbd, sr_hse 等任务上取得了显著的提升。具体而言, sr_p53 任务提升了 4.5%, nr_ahr 任务提升了 1.7%, nr_er 任务提升了 3.1%, nr_ar 任务提升了 7.5%, sr_are 任务提升了 4.8%, nr_ar_lbd 任务提升了 2.6%, nr_er_lbd 任务提升了 1.5%, sr_hse 提升了 3.3%。而在 nr_aromatase, nr_ppar_gamma, sr_atad5, sr_mmp 等任务上的表现不如预期。

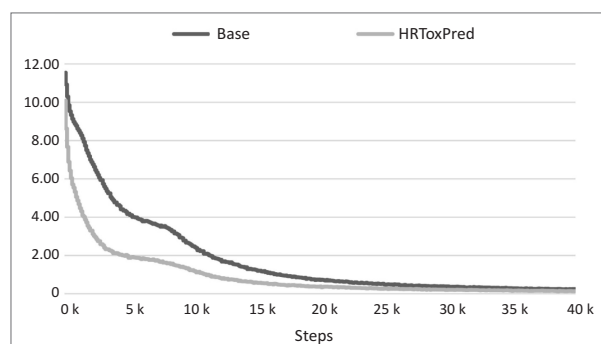


图 4 Base 和 HRToxPred 预训练 Loss 曲线

3.7 对比实验分析

为了检验 HRToxPred 的有效性,将 HRToxPred 与 Chemberta 和 Mol-BERT 模型进行了比较。表 6 报告了各个模型 Tox21 数据集 12 个通道上的 AUC 分数。HRToxPred 在 sr_p53, nr_ahr, nr_er, nr_ar, sr_are, nr_ar_lbd, nr_aromatase, nr_er_lbd, sr_hse 等任务上取得了最佳的表现。同时,如图 5 所示, HRToxPred 在 Tox21 的雷达图中拥有最大的面积。

表 5 消融实验结果比较(ROC-AUC ↑)

模型	sr_p53	nr_ahr	nr_er	nr_ar	sr_are	nr_ar_lbd	nr_aromatase	nr_er_lbd	nr_ppar_gamma	sr_atad5	sr_hse	sr_mmp
Base	0.818	0.874	0.758	0.756	0.775	0.855	0.842	0.819	0.868	0.872	0.792	0.907
AddHS	0.839	0.882	0.762	0.761	0.793	0.872	0.885	0.826	0.869	0.886	0.808	0.939
AddRadius	0.847	0.879	0.764	0.752	0.789	0.857	0.841	0.825	0.804	0.853	0.799	0.909
HRToxPred	0.863	0.891	0.787	0.831	0.823	0.881	0.851	0.834	0.794	0.850	0.825	0.889

表 6 对比实验结果比较(ROC-AUC ↑)

模型	sr_p53	nr_ahr	nr_er	nr_ar	sr_are	nr_ar_lbd	nr_aromatase	nr_er_lbd	nr_ppar_gamma	sr_atad5	sr_hse	sr_mmp
Mol-BERT	0.818	0.874	0.758	0.756	0.775	0.855	0.842	0.812	0.868	0.872	0.792	0.907
Chemberta	0.723	0.808	0.714	0.736	0.703	0.657	0.596	0.722	0.738	0.642	0.715	0.734
HRToxPred	0.863	0.891	0.787	0.831	0.823	0.881	0.851	0.834	0.794	0.850	0.825	0.889

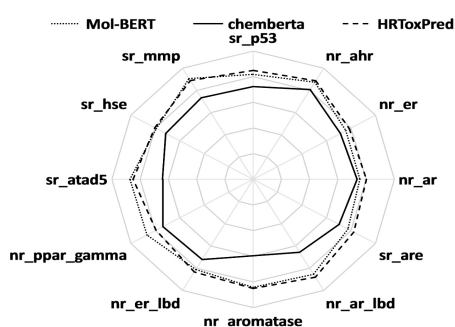


图 5 Tox21 各任务 AUC 分数雷达图

4 结束语

该文提出了引入摩根指纹半径以及显式引入氢原子的知识增强方法,并预训练了一个原子知识增强的分子毒性预测模型 HRToxPred。该模型与现有的基于序列的分子毒性预测模型有两个方面的优势:一方面是显式地引入了氢原子的知识,使得 MFI 字典规模更小,进而减少了模型参数量并提升了毒性预测的性能;另一方面是额外引入了摩根指纹半径的知识,增强了 MFI 序列的顺序特征。从对比实验和消融实验的结果可见,这两方面能使模型性能有了显著提升。

虽然该方法在一些毒性预测任务数据集上展现了良好的性能,但仍有一些局限性。从消融实验的结果来看,HRToxPred 在 Tox21 数据集中 nr_aromatase, nr_ppar_gamma, sr_atad5, sr_mmp 等任务上的表现不如预期,后续工作会结合深度学习理论和结构化学的知识来详细分析引入氢原子及半径对不同毒性预测任务的影响。此外,根据化学的领域知识可知,分子中的官能团对分子的特性有较大影响^[20],因此后续工作会考虑引入相关领域知识。

参考文献:

- [1] 梁礼,邓成龙,张艳敏,等.人工智能在药物发现中的应用与挑战[J].药学进展,2020,44(1):18-27.
- [2] WEININGER D. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules[J]. Journal of Chemical Information and Computer Sciences, 1988, 28(1):31-36.
- [3] CHITHRANANDA S, GRAND G, RAMSUNDAR B. Chemberta: large-scale self-supervised pretraining for molecular property prediction[J]. arXiv:2010.09885, 2020.
- [4] LI J, JIANG X. Mol-bert: an effective molecular representation with bert for molecular property prediction[J]. Wireless Communications and Mobile Computing, 2021, 2021:1-7.
- [5] GLEN R C, BENDER A, ARNBY C H, et al. Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to ADME[J]. IDrugs, 2006, 9(3):199-204.
- [6] LANDRUM G. Rdkit documentation[J]. Release, 2013, 1:1-79.
- [7] SENNRICH R, HADDOW B, BIRCH A. Neural machine translation of rare words with subword units[C]//Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers). Berlin: Association for Computational Linguistics, 2016:1715-1725.
- [8] LI Piji, ZHANG Haisong, LIU Xiaojiang, et al. Rigid formats controlled text generation[C]//Proceedings of the 58th annual meeting of the association for computational linguistics. [s. l.]: Association for Computational Linguistics, 2020:742-751.
- [9] KIM S, THIESSEN P A, BOLTON E E, et al. PubChem substance and compound databases[J]. Nucleic Acids Research, 2016, 44(D1):D1202-D1213.
- [10] RICHARD A M, HUANG R, WAIDYANATHA S, et al. The Tox21 10K compound library: collaborative chemistry advancing toxicology[J]. Chemical Research in Toxicology, 2020, 34(2):189-216.
- [11] JEON W, KIM D. FP2VEC: a new molecular featurizer for learning molecular properties[J]. Bioinformatics, 2019, 35(23):4979-4985.
- [12] JAEGER S, FULLE S, TURK S. Mol2vec: unsupervised machine learning approach with chemical intuition[J]. Journal of Chemical Information and Modeling, 2018, 58(1):27-35.
- [13] CHURCH K W. Word2Vec[J]. Natural Language Engineering, 2017, 23(1):155-162.
- [14] LIU Y, OTT M, GOYAL N, et al. Roberta: a robustly optimized bert pretraining approach[J]. arXiv:1907.11692, 2019.
- [15] DEVLIN J, CHANG M W, LEE K, et al. Bert: pre-training of deep bidirectional transformers for language understanding[J]. arXiv:1810.04805, 2018.
- [16] LANDRUM G. RDKit: a software suite for cheminformatics, computational chemistry, and predictive modeling[EB/OL]. 2013. http://www.rdkit.org/RDKit_Overview.pdf.
- [17] RAMSUNDAR B, EASTMAN P, WALTERS P, et al. Deep learning for the life sciences: applying deep learning to genomics, microscopy, drug discovery, and more[M]. [s. l.]: O'Reilly Media, 2019.
- [18] HANLEY J A, MCNEIL B J. A method of comparing the areas under receiver operating characteristic curves derived from the same cases[J]. Radiology, 1983, 148(3):839-843.
- [19] ZHAO Z, CHEN H, ZHANG J, et al. UER: an open-source toolkit for pre-training models[J]. EMNLP-IJCNLP, 2019, 2019:241.
- [20] CHEN F, PARK J, PARK J. A hypergraph convolutional neural network for molecular properties prediction using functional group[J]. arXiv:2106.01028, 2021.