frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 29 January 2025
DOI 10.3389/fgene.2024.1483490

OPEN ACCESS

EDITED BY
Shoba Ranganathan,
Macquarie University, Australia

REVIEWED BY
Yuansheng Liu,
Hunan University, China
Tsung Fei Khang,
University of Malaya, Malaysia

*CORRESPONDENCE
Naveed Ahmed Azam,
naveedazam@qau.edu.pk

RECEIVED 20 August 2024
ACCEPTED 30 December 2024
PUBLISHED 29 January 2025

CITATION
Zhu J, Azam NA, Cao S, Ido R, Haraguchi K, Zhao L, Nagamochi H and Akutsu T (2025) Quadratic descriptors and reduction methods in a two-layered model for compound inference.
Front. Genet. 15:1483490.
doi: 10.3389/fgene.2024.1483490

Quadratic descriptors and reduction methods in a two-layered model for compound inference

Jianshen Zhu[1], Naveed Ahmed Azam[2]*, Shengjuan Cao[1], Ryota Ido[1], Kazuya Haraguchi[1], Liang Zhao[3], Hiroshi Nagamochi[1] and Tatsuya Akutsu[4]

[1]Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, Japan, [2]Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan, [3]Graduate School of Advanced Integrated Studies in Human Survivability (Shishu-Kan), Kyoto University, Kyoto, Japan, [4]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Japan

Compound inference models are crucial for discovering novel drugs in bioinformatics and chemo-informatics. These models rely heavily on useful descriptors of chemical compounds that effectively capture important information about the underlying compounds for constructing accurate prediction functions. In this article, we introduce quadratic descriptors, the products of two graph-theoretic descriptors, to enhance the learning performance of a novel two-layered compound inference model. A mixed-integer linear programming formulation is designed to approximate these quadratic descriptors for inferring desired compounds with the two-layered model. Furthermore, we introduce different methods to reduce descriptors, aiming to avoid computational complexity and overfitting issues during the learning process caused by the large number of quadratic descriptors. Experimental results show that for 32 chemical properties of monomers and 10 chemical properties of polymers, the prediction functions constructed by the proposed method achieved high test coefficients of determination. Furthermore, our method inferred chemical compounds in a time ranging from a few seconds to approximately 60 s. These results indicate a strong correlation between the properties of chemical graphs and their quadratic graph-theoretic descriptors.

KEYWORDS

machine learning, integer programming, chemo-informatics, materials informatics, QSAR/QSPR, molecular design

1 Introduction

In recent years, extensive studies have been done on the design of novel molecules using various machine learning techniques (Lo et al., 2018; Tetko and Engkvist, 2020; Gartner III et al., 2024). Computational molecular design also has a long history in the field of chemo-informatics and has been studied under the names of *quantitative structure-activity relationship* (QSAR) (Cherkasov et al., 2014) and *inverse quantitative structure-activity relationship* (inverse QSAR) (Miyao et al., 2016; Ikebata et al., 2017; Rupakheti et al., 2015). This design problem has also become an important topic in both bioinformatics and machine learning.

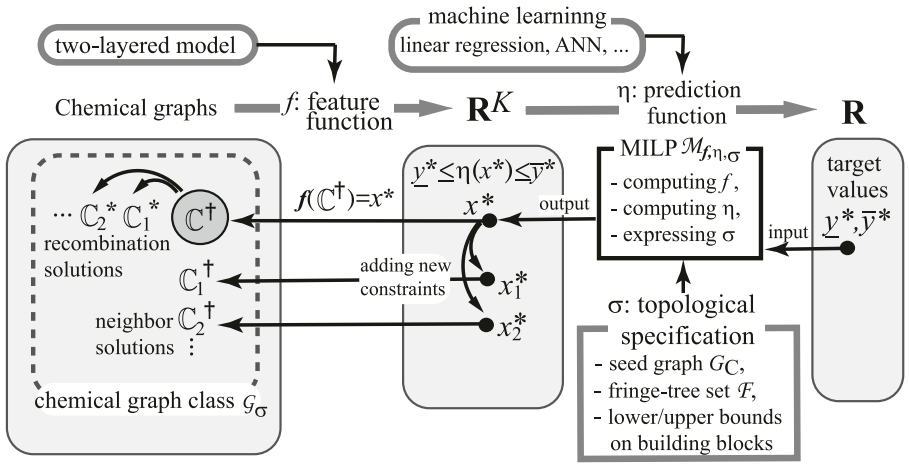

FIGURE 1
An illustration of inferring desired chemical graphs $\mathbb{C} \in \mathcal{G}_\sigma$ with $\underline{y}^* \le \eta(f(\mathbb{C})) \le \bar{y}^*$.

The purpose of QSAR is to predict chemical activities from given chemical structures (Cherkasov et al., 2014). In most QSAR studies, a chemical structure is represented as a vector of real numbers called *features* or *descriptors*, and then a prediction function is applied to the vector, where a chemical structure is given as a *chemical graph*, which is defined by an undirected graph with an assignment of chemical elements to vertices and an assignment of bond multiplicities to edges. A prediction function is usually obtained from existing structure–activity relationship data. To this end, various regression-based methods have been utilized in traditional QSAR studies, whereas machine learning-based methods, including artificial neural network (ANN)-based methods, have recently been utilized (Ghasemi et al., 2018; Kim et al., 2021; Catacutan et al., 2024).

Conversely, the purpose of inverse QSAR is to predict chemical structures from given chemical activities (Miyao et al., 2016; Ikebata et al., 2017; Rupakheti et al., 2015; Miyao and Funatsu, 2024), where additional constraints may often be imposed to effectively restrict the possible structures. In traditional inverse QSAR, a feature vector is computed by applying some optimization or sampling method on the prediction function obtained by standard QSAR, and then chemical structures are reconstructed from the feature vector. However, the reconstruction itself is quite difficult due to the vast number of possible chemical graphs (Bohacek et al., 1996). In fact, inferring a chemical graph from a given feature vector, except for some simple cases, is an NP-hard problem (Akutsu et al., 2012). Due to this inherent difficulty, most existing methods employ heuristic methods for the reconstruction of chemical structures and thus do not guarantee optimal or exact solutions. On the other hand, one of the advantages of ANNs is that generative models, such as autoencoders and generative adversarial networks, are available. Furthermore, graph-structured data can be directly handled by using graph convolutional networks (Kipf and Welling, 2016). Therefore, it is reasonable to try to apply ANNs to inverse QSAR (Xiong et al., 2021). Various ANN models, including recurrent networks, autoencoders, generative networks, and invertible flow models, have been applied in this context (Segler et al., 2018; Yang et al., 2017; Gómez-Bombarelli et al., 2018; Kusner et al., 2017; De Cao and Kipf, 2018; Madhawa et al., 2019; Shi et al., 2020). However, the optimality or exactness of the solutions provided by these methods is not yet guaranteed.

A novel two-phase framework based on mixed-integer linear programming (MILP) and machine learning methods has been developed to infer chemical graphs (Shi et al., 2021; Zhu et al., 2022b; Ido et al., 2021; Azam et al., 2021a). The first phase constructs a prediction function η for a chemical property, and the second phase infers a chemical graph with a target value of the property based on the function η. For a chemical property π, we define $\mathcal{C}_\pi$ as a data set of chemical graphs such that the observed value $a(\mathbb{C})$ of property π for every chemical graph $\mathbb{C} \in \mathcal{C}_\pi$ is available.

In the first phase, we introduce a feature function $f: \mathcal{G} \to \mathbb{R}^K$ for a positive integer K, where the descriptors of a chemical graph are defined based on local graph structures by using a *two-layered model* (Shi et al., 2021) so that the inverse of f can be modeled by MILP in the second phase. The prediction function aims to produce an output $y = \eta(x) \in \mathbb{R}$ based on the feature vector $x = f(\mathbb{C}) \in \mathbb{R}^K$ for each $\mathbb{C} \in \mathcal{C}_\pi$. This output serves as a predicted value for the real value $a(\mathbb{C})$.

The task of the second phase is to infer desired chemical graphs. This phase consists of three steps. For a given set of rules called *topological specification* σ and a range $[\underline{y}^*, \bar{y}^*]$ of the target property value, the aim of the first step is to infer chemical graphs $\mathbb{C}^*$ that satisfy the rules σ and $\underline{y}^* \le \eta(f(\mathbb{C}^*)) \le \bar{y}^*$ (see Figure 1 for an illustration). For this, we formulate an MILP $\mathcal{M}_{f,\eta,\sigma}$ that represents (i) the computation process of $x := f(\mathbb{C})$ from a chemical graph $\mathbb{C}$ in the feature function f; (ii) that of $y := \eta(x)$ from a vector $x \in \mathbb{R}^K$ in the prediction function η; and (iii) the constraint $\mathbb{C} \in \mathcal{G}_\sigma$, where $\mathcal{G}_\sigma$ denotes the set of all chemical graphs that satisfy the rules in σ. Given an interval with $\underline{y}^*, \bar{y}^* \in \mathbb{R}$, we solve the MILP $\mathcal{M}_{f,\eta,\sigma}$ to find a feature vector $x^* \in \mathbb{R}^K$ and a chemical graph $\mathbb{C}^\dagger \in \mathcal{G}_\sigma$ such that $f(\mathbb{C}^\dagger) = x^*$ and $\underline{y}^* \le \eta(x^*) \le \bar{y}^*$ (if the MILP instance is infeasible, this suggests that $\mathcal{G}_\sigma$ does not contain such a desired chemical graph). See Zhu et al. (2022a) for a full description of the framework.

In the second and third steps of the process, we employ different techniques to generate additional desired chemical graphs based on

TABLE 1 Results of setting data sets for monomers.

π	Λ	$\lvert\mathcal{C}_\pi\rvert$	$\underline{n}, \bar{n}$	$\underline{a}, \bar{a}$	$\lvert\Gamma\rvert$	$\lvert\mathcal{F}\rvert$	K_1
Biological half life (BHL) (PubChem, 2022)	Λ_7	514	5, 36	0.03, 732.99	26	101	166
Boiling point (BP) (PubChem, 2022)	Λ_2	370	4, 67	−11.7, 470.0	22	130	184
Boiling point (BP) (PubChem, 2022)	Λ_7	444	4, 67	−11.7, 470.0	26	163	230
Critical pressure (CP) (PubChem, 2022)	Λ_5	131	4, 63	4.7×10^{-6}, 5.52	8	79	119
Critical temperature (CT) (PubChem, 2022)	Λ_2	125	4, 63	56.1, 3607.5	8	76	113
Critical temperature (CT) (PubChem, 2022)	Λ_5	132	4, 63	56.1, 3607.5	8	81	121
Dissociation constants (DC) (PubChem, 2022)	Λ_2	141	5, 44	0.5, 17.11	20	62	111
Dissociation constants (DC) (PubChem, 2022)	Λ_7	161	5, 44	0.5, 17.11	25	69	130
Entropy (ET) (Duchowicz et al., 2002)	Λ_7	17	5, 12	64.34, 96.21	5	17	53
Flash point in closed cup (FP) (PubChem, 2022)	Λ_2	368	4, 67	−82.99, 300.0	20	131	183
Flash point in closed cup (FP) (PubChem, 2022)	Λ_7	424	4, 67	−82.99, 300.0	25	161	229
Flammable limits lower of organics (FLMLO) (Yuan et al., 2019)	Λ_{16}	1046	1, 49	0.185, 4.3	34	282	376
Heat of vaporization (HV) (PubChem, 2022)	Λ_2	94	4, 16	19.12, 210.3	12	63	105
Kovats retention index (KOV) (Jalali-Heravi and Fatemi, 2001)	Λ_1	52	11, 16	1422.0, 1919.0	9	33	64
Octanol/water partition coefficient (KOW) (PubChem, 2022)	Λ_2	684	4, 58	−7.5, 15.6	25	166	223
Octanol/water partition coefficient (KOW) (PubChem, 2022)	Λ_8	899	4, 69	−7.5, 15.6	37	219	303
Lipophilicity (LP) (Xiao, 2017)	Λ_2	615	6, 60	−3.62, 6.84	32	116	186
Lipophilicity (LP) (Xiao, 2017)	Λ_8	936	6, 74	−3.62, 6.84	44	136	231
Melting point (MP) (PubChem, 2022)	Λ_2	467	4, 122	−185.33, 300.0	23	142	197
Melting point (MP) (PubChem, 2022)	Λ_8	577	4, 122	−185.33, 300.0	32	176	255
Optical rotation (OPTR) (PubChem, 2022)	Λ_2	147	5, 44	−117.0, 165.0	21	55	107
Optical rotation (OPTR) (PubChem, 2022)	Λ_4	157	5, 69	−117.0, 165.0	25	62	123
Refractive index of trees (RFIDT) (PubChem, 2022)	Λ_{10}	191	4, 26	0.919, 1.613	17	115	168
Solubility (SL) (MoleculeNet, 2022)	Λ_2	673	4, 55	−9.332, 1.11	27	154	217
Solubility (SL) (MoleculeNet, 2022)	Λ_8	915	4, 55	−11.6, 1.11	42	207	300
Surface tension (SFT) (Goussard et al., 2017)	Λ_3	247	5, 33	12.3, 45.1	11	91	128
Viscosity (VIS) (Goussard et al., 2017)	Λ_3	282	5, 36	−0.64, 1.63	12	88	126
Energy of highest-occupied molecular orbital (HOMO) (MoleculeNet, 2022)	Λ_9	977	6, 9	−0.3335, −0.1583	59	190	297
Energy of lowest-unoccupied molecular orbital (LUMO) (MoleculeNet, 2022)	Λ_9	977	6, 9	−0.1144, 0.1026	59	190	297
The energy difference between HOMO and LUMO (GAP) (MoleculeNet, 2022)	Λ_9	977	6, 9	0.1324, 0.4117	59	190	297
Isotropic polarizability (ALPHA) (MoleculeNet, 2022)	Λ_9	977	6, 9	50.9, 99.6	59	190	297
Heat capacity at 298.15 K (CV) (MoleculeNet, 2022)	Λ_9	977	6, 9	19.2, 44.0	59	190	297
Electric dipole moment (MU) (MoleculeNet, 2022)	Λ_9	977	6, 9	0.04, 6.897	59	190	297

the initial solution $\mathbb{C}^\dagger$. These techniques include a dynamic programming algorithm (Azam et al., 2021b) and a neighbor search method (Azam et al., 2021a). The dynamic programming algorithm operates by decomposing $\mathbb{C}^\dagger$ into trees and generating their isomers. These isomers are then combined to produce a set of isomers $\mathbb{C}^\star$ that belong to the desired chemical graph space $\mathcal{G}_\sigma$. These isomers are referred to as the *recombination solutions* of $\mathbb{C}^\dagger$. The idea of the neighbor search method is to generate new desired chemical graphs $\mathbb{C}_i^\dagger \in \mathcal{G}_\sigma$ that have a slightly different feature vector than that of $\mathbb{C}^\dagger$. For this purpose, we solve an augmented MILP obtained by $\mathcal{M}_{f,\eta,\sigma}$ with an additional set of linear constraints. The resulting graphs obtained through this process are referred to as the *neighbor solutions* of $\mathbb{C}^\dagger$.

Contribution: The feature vector $x = f(\mathbb{C})$ of a chemical graph $\mathbb{C}$ in this framework consists of the descriptors $x(i)$, $1 \le i \le K$ that extract the local information of the interior and exterior parts of the graph obtained from the two-layered model. These simple descriptors play a

TABLE 2 Results of setting data sets for polymers.

π	Λ	$\lvert\mathcal{C}_\pi\rvert$	$\underline{n}, \bar{n}$	$\underline{a}, \bar{a}$	$\lvert\Gamma\rvert$	$\lvert\mathcal{F}\rvert$	K_1
Experimental amorphous density (AMD) (Bicerano, 2002)	Λ_2	86	4, 45	0.838, 1.34	16	25	83
Experimental amorphous density (AMD) (Bicerano, 2002)	Λ_{13}	93	4, 45	0.838, 1.45	18	30	94
Characteristic ratio (CHAR) (Bicerano, 2002)	Λ_2	30	4, 18	3.7, 15.9	15	17	68
Characteristic ratio (CHAR) (Bicerano, 2002)	Λ_{12}	32	4, 18	3.7, 15.9	15	18	71
Characteristic ratio (CHAR) (Bicerano, 2002)	Λ_6	35	4, 18	3.7, 15.9	18	21	83
Dielectric constant (DIEC) (Bicerano, 2002)	Λ_{12}	36	4, 22	2.13, 3.4	11	18	67
Dissipation factor (DISF) (Bicerano, 2002)	Λ_{13}	132	4, 45	7×10^{-5}, 0.07	15	18	78
Permittivity (PRM) (Bicerano, 2002)	Λ_2	112	4, 45	2.23, 4.91	14	15	69
Permittivity (PRM) (Bicerano, 2002)	Λ_{13}	132	4, 45	2.23, 4.91	15	18	78
Refractive index of polymers (RFIDP) (Bicerano, 2002)	Λ_{11}	92	4, 29	0.4899, 1.683	15	35	96
Refractive index of polymers (RFIDP) (Bicerano, 2002)	Λ_{14}	125	4, 29	0.4899, 1.683	19	50	124
Refractive index of polymers (RFIDP) (Bicerano, 2002)	Λ_{15}	135	4, 29	0.4899, 1.71	23	56	144
Glass transition (TG) (Bicerano, 2002)	Λ_2	204	4, 58	171, 673	19	36	101
Glass transition (TG) (Bicerano, 2002)	Λ_7	232	4, 58	171, 673	21	43	118

crucial role in deriving compact MILP formulations for the inference of a desired chemical graph. However, for certain chemical properties, the prediction functions constructed based on the feature function f could not achieve evaluation scores in the acceptable range. To improve the evaluation score, we introduce new *quadratic descriptors* $x(i)x(j)$ (or $x(i)(1-x(j))$). This drastically increases the number of descriptors, which would take extra running time for learning or lead to overfitting of the data set. Moreover, computed quadratic descriptors cannot be directly formulated as a set of linear constraints in the original MILP. For this, we introduce a method of reducing a set of descriptors into a smaller set that delivers a prediction function with a higher performance. We also design an MILP formulation to represent a quadratic term $x(i)x(j)$. Based on the same MILP $\mathcal{M}_{f,\eta,\sigma}$ formulation proposed by Zhu et al. (2022b), we implemented the framework to treat the feature function with quadratic descriptors. From the results of our computational experiments on more than 40 chemical properties, we observe that our new method of utilizing quadratic descriptors has improved the performance of a prediction function for many chemical properties.

2 Quadratic descriptors and their approximation in an MILP

In the framework with the two-layered model (Shi et al., 2021), the feature vector consists of graph-theoretic descriptors that are mainly the frequencies of atoms, bonds, and edge configurations in the interior and exterior of the chemical compounds. See Zhu et al. (2022a) for the details of the interior and exterior of chemical compounds and all the descriptors $x(1), x(2), \ldots, x(K_1)$, which are called *linear descriptors*, respectively. There are some chemical properties for which the performance of a prediction function constructed with this feature vector remains rather low. To improve the learning performance in the two-layered model, we introduce quadratic terms $x(i)x(j)$ (or $x(i)(1-x(j))$) and $1 \le i \le j \le K_1$ as a new descriptor, where we assume that each $x(i)$ is normalized between 0 and 1 by using the min-max normalization. Note that computing quadratic descriptors cannot be directly formulated as a set of linear constraints in the original MILP used in the two-layered model. Therefore, this section introduces an MILP formulation that approximates the product of two descriptors in a MILP.

Given two real values x and y with $0 \le x \le 1$ and $0 \le y \le 1$, the process of computing the product $z = xy$ can be approximately formulated as the following MILP. First, regard $(2^{p+1}-1)x$ as an integer with a binary expression of $p+1$ bits, where $x^{(j)} \in [0,1]$ denotes the value of the j-th bit. Then, compute $y \cdot x^{(j)}$, which becomes the j-th bit $z^{(j)}$ of $(2^{p+1}-1)z$.

Constants:

- x, y: reals with $0 \le x, y \le 1$;
- p: a positive integer;

variables:

- z, $z^{(j)}$, $j \in [0,p]$: reals with $0 \le z, z^{(j)} \le 1$;
- $x^{(j)} \in [0,1]$, $j \in [0,p]$: binary variables;

constraints:

$$\sum_{j\in[0,p]} 2^j x^{(j)} - 1 \le (2^{p+1}-1)x \le \sum_{j\in[0,p]} 2^j x^{(j)},$$

$$z^{(j)} \le x^{(j)}, \quad j \in [0,p],$$

$$y - \left(1 - x^{(j)}\right) \le z^{(j)} \le y + \left(1 - x^{(j)}\right), \quad j \in [0,p],$$

$$z = \frac{1}{2^{p+1}-1} \sum_{j\in[0,p]} 2^j z^{(j)}.$$

Note that the necessary number of integer variables for computing xy for one pair of x and y is p. In this article, we set $p := 6$ in our computational experiment. The relative error by $p = 6$ in the above method is at most $\frac{1}{2^{p+1}-1} = 1/127$, which is approximately 0.8%.

TABLE 3 Results of constructing prediction functions for monomers.

π	Λ	LLR	ANN	ALR	R-MLR	K_1^*, K_2^*
BHL	Λ_7	0.483	0.622	0.265	*0.659	0, 27
BP	Λ_2	0.599	0.765	0.816	*0.935	1, 59
BP	Λ_7	0.663	0.720	0.832	*0.899	0, 38
CP	Λ_5	0.555	0.727	0.690	*0.841	0, 67
CT	Λ_2	0.037	0.357	0.900	*0.937	1, 47
CT	Λ_5	0.048	0.357	*0.895	0.860	0, 13
DC	Λ_2	0.489	0.651	0.488	*0.908	0, 58
DC	Λ_7	0.574	0.622	0.602	*0.829	0, 26
ET	Λ_7	0.132	0.479	0.464	*0.996	0, 13
FP	Λ_2	0.589	0.746	0.719	*0.899	0, 42
FP	Λ_7	0.571	0.745	0.684	*0.846	0, 32
FIMLO	Λ_{16}	0.819	0.928	0.604	*0.949	0, 77
HV	Λ_2	0.864	0.778	0.816	*0.970	0, 22
KOV	Λ_1	0.677	0.727	0.838	*0.953	2, 19
KOW	Λ_2	0.953	0.952	0.964	*0.967	0, 55
KOW	Λ_8	0.927	0.937	*0.952	0.950	0, 64
LP	Λ_2	0.856	0.867	0.844	*0.928	0, 89
LP	Λ_8	0.840	0.859	0.807	*0.914	0, 109
MP	Λ_2	0.810	0.800	0.831	*0.873	0, 51
MP	Λ_8	0.810	0.820	0.807	*0.898	0, 58
OPTR	Λ_2	0.825	0.918	0.876	*0.970	0, 85
OPTR	Λ_4	0.825	0.878	0.870	*0.970	0, 69
RFIDT	Λ_{10}	0.000	0.453	0.425	*0.775	0, 43
SL	Λ_2	0.808	0.848	0.784	*0.894	0, 82
SL	Λ_8	0.808	0.861	0.828	*0.897	0, 74
SFT	Λ_3	0.927	0.859	0.847	*0.941	0, 36
VIS	Λ_3	0.893	0.929	0.911	*0.973	0, 43
HOMO	Λ_9	*0.841	0.689	0.689	0.804	0, 87
LUMO	Λ_9	0.841	0.860	0.833	*0.920	0, 102
GAP	Λ_9	0.784	0.795	0.755	*0.876	0, 83
ALPHA	Λ_9	0.961	0.888	0.953	*0.980	0, 104
CV	Λ_9	0.970	0.911	0.966	*0.978	0, 83
MU	Λ_9	0.367	0.409	0.403	*0.645	0, 112

3 Methods for reducing descriptors

The number of descriptors in the two-layered model increases drastically due to the quadratic descriptors. More precisely, if $D_\pi^{(1)} := \{x(k) \mid k \in [1, K_1]\}$ denotes the set of linear descriptors, and $D_\pi^{(2)} := \{x(i)x(j) \mid 1 \le i \le j \le K_1\} \cup \{x(i)(1 - x(j)) \mid 1 \le i, j \le K_1\}$ denotes the set of quadratic descriptors constructed over a data set for a property π, then the total number of descriptors are $K_1 + (3(K_1)^2 + K_1)/2$. This large number of descriptors would increase the running time to construct prediction functions or lead to overfitting of the data set. To address these issues, we propose methods for reducing descriptors in this section.

TABLE 4 Results of constructing prediction functions for polymers. The negative value shows that the respective model is arbitrarily worse.

π	Λ	LLR	ANN	ALR	R-MLR	K_1^*, K_2^*
AMD	Λ_2	0.914	0.885	*0.933	0.906	0, 5
AMD	Λ_{13}	0.918	0.824	0.917	*0.953	0, 6
CHAR	Λ_2	0.210	0.642	0.863	*0.938	0, 10
CHAR	Λ_{12}	0.088	0.640	0.835	*0.924	0, 9
CHAR	Λ_6	−0.073	0.527	0.766	*0.950	0, 12
DEIC	Λ_{12}	0.761	0.641	0.918	*0.956	3, 41
DISF	Λ_{13}	0.623	0.801	0.308	*0.906	1, 23
PRM	Λ_2	0.801	0.801	0.505	*0.967	0, 26
PRM	Λ_{13}	0.784	0.735	0.489	*0.977	0, 34
RFIDP	Λ_{11}	0.104	0.423	0.853	*0.962	2, 52
RFIDP	Λ_{14}	0.373	0.560	0.848	*0.953	2, 43
RFIDP	Λ_{15}	0.346	0.492	0.883	*0.947	5, 53
TG	Λ_2	0.902	0.883	0.923	*0.958	1, 33
TG	Λ_7	0.894	0.860	0.927	*0.957	0, 32

Let $\mathcal{C}$ be a given set of chemical compounds, D be a descriptor set obtained using the two-layered model, and $K^* \in [1, |D|]$ be the size of a set of selected descriptors.

For given $\mathcal{C}$, D and a real number $\lambda > 0$, we denote by Des-set-LLR$(\mathcal{C}, D, \lambda)$ the subset S of D such that $w(d) = 0$ for $d \in S$ and the hyperplane (w, b) constructed by the LLR procedure LLR$(\mathcal{C}, D, \lambda)$.

3.1 A method based on lasso linear regression

Because the lasso linear regression finds some number of descriptors $d \in D$ with $w(d) = 0$ in the output hyperplane (w, b), we can reduce a given set of descriptors by applying the lasso linear regression repeatedly. Choose parameters $c_{\max}$ and $d_{\max}$ so that LLR$(\mathcal{C}, D, \lambda)$ can be executed in a reasonable running time when $|\mathcal{C}| \le c_{\max}$ and $|D| \le d_{\max}$. Let $\tilde{K} \in [1, |D|]$ be an integer for the number of descriptors that we choose from a given set D of descriptors.

LLR-Reduce$(\mathcal{C}, D)$:

Input: A data set $\mathcal{C}$ and a set D of descriptors;
Output: A subset $\tilde{D} \subseteq D$ with $|\tilde{D}| = \tilde{K}$.
Initialize $D' := D$;
while $|D'| > \tilde{K}$ **do**
  Partition D' randomly into disjoint subsets $D_1, D_2, \ldots, D_p$ such that
    $|D_i| \le d_{\max}$ for each i;
  for each $i = 1, 2, \ldots, p$ **do**
    Choose a subset $\mathcal{C}_i$ with $|\mathcal{C}_i| = \min\{c_{\max}, |\mathcal{C}|\}$ of $\mathcal{C}$ randomly;
    $D_i' :=$ Des-set-LLR$(\mathcal{C}_i, D_i, \lambda)$ for some $\lambda > 0$
  end for;
$D' := D_1' \cup D_2' \cup \cdots \cup D_p'$
end while;
Output $\tilde{D} := D'$ after adding to D' extra $\tilde{K} - |D'|$ descriptors from the previous
set D' when $|D'| < \tilde{K}$ by using the K-best method.

In this article, we set $c_{\max} := 200$, $d_{\max} := 200$, $\tilde{K} := 5000$ and $w(d) := 0$ if $|w(d)| \le 10^{-6}$ in our computational experiment.

3.2 A method based on the backward stepwise procedure

A backward stepwise procedure (Draper and Smith, 1998) reduces the number of descriptors one by one, choosing the one removal that maximizes the learning performance and outputs a subset with the maximum learning performance among all subsets during the reduction iteration.

For a subset $S \subseteq D$ and a positive integer p, let $R^2_{CV,MLR}(\mathcal{C}, S, p)$ denote the median of the coefficient of determination R^2 test score of a prediction function $\eta_{w,b}$ by MLR$(\mathcal{C}, S)$ constructed during p times 5-fold cross-validations. We define a performance evaluation function $g_p: 2^D \to \mathbb{R}$ for an integer $p \ge 1$ such that $g_p(S) = R^2_{CV,MLR}(\mathcal{C}, S, p)$. The backward stepwise procedure with this function g_p is described as follows.

BS-Reduce$(\mathcal{C}, D, p)$:

Input: A data set $\mathcal{C}$, a set D of descriptors, an integer $p \ge 1$, and a performance evaluation function $g_p: 2^D \to \mathbb{R}$ defined above;
Output: A subset $D^* \subseteq D$.
Compute $\ell_{\text{best}} := g_p(D)$; Initialize $D_{\text{best}} := D' := D$;
while $D' \neq \varnothing$ **do**
  Compute $\ell(d) := g_p(D' \backslash \{d\})$ for each descriptor $d \in D'$;

10.3389/fgene.2024.1483490

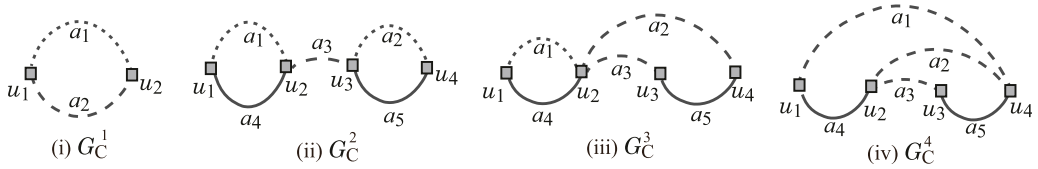

FIGURE 2
(i) Seed graph G_C^1 for I_b^1 and I_d; (ii) seed graph G_C^2 for I_b^2; (iii) seed graph G_C^3 for I_b^3; (iv) seed graph G_C^4 for I_b^4.

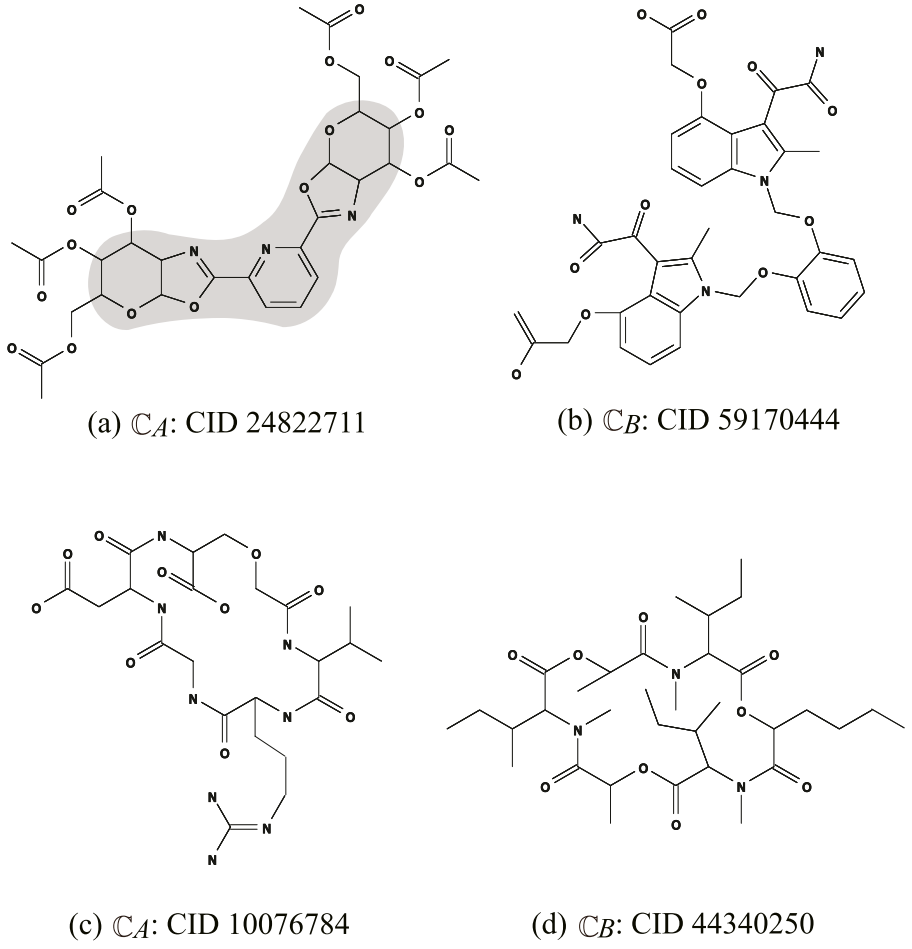

FIGURE 3
An illustration of chemical compounds for instances I_c and I_d: **(A)** $\mathbb{C}_A$: CID 24822711; **(B)** $\mathbb{C}_B$: CID 59170444; **(C)** $\mathbb{C}_A$: CID 10076784; **(D)** $\mathbb{C}_B$: CID 44340250, where hydrogens are omitted.

Set $d^\star \in D'$ to be a descriptor that maximizes $\ell(d)$ over all $d \in D'$;

Update: $D' := D' \backslash \{d^\star\}$;

if $\ell(d^\star) > \ell_{\text{best}}$ **then** update $D_{\text{best}} := D'$ and $\ell_{\text{best}} := \ell(d^\star)$ **end if**

end while;

Output $D^\star := D_{\text{best}}$.

Based on the lasso linear regression and the backward stepwise procedure, we design the following method for choosing a subset $D^\star$ of a given set D of descriptors. We are given a set A of 17 real numbers and a set $B(a)$ of 16 real numbers close to each number $a \in A$. The method first chooses a best parameter $\lambda_{\text{best}} \in A$ to construct a prediction function by LLR and then chooses a subset $D_i \subseteq D$ for each $\lambda_i \in B(\lambda_{\text{best}})$ by the backward stepwise procedure. The procedure takes $\mathcal{O}(|D|^2)$ iterations, which may take a large amount of running time. We introduce an upper bound $s_{\max}$ on the size of an input descriptor set D for the backward stepwise procedure. Let p_1, p_2, and p_3 be integer parameters that control the number of executions of cross-validations to evaluate the learning performance in the method.

Select-Des-set $(\mathcal{C}, D)$:

TABLE 5 Results of inferring a chemical graph $\mathbb{C}^{\dagger}$ and generating recombination solutions for BP with Λ_7.

inst.	n_{LB}	$\underline{y}^*, \bar{y}^*$	#v	#c	l-time	n	n^{int}	η	D-time	$\mathbb{C}$-LB	#$\mathbb{C}$
I_{a}	30	225, 235	10,502	10,240	4.29	49	26	233.92	0.072	3	3
I_{b}^1	35	285, 295	10,507	7,793	2.27	35	10	286.52	0.034	6	6
I_{b}^2	45	365, 375	13,000	10,913	11.9	49	25	370.70	0.14	3202	100
I_{b}^3	45	305, 315	12,788	10,920	7.07	48	25	309.39	0.22	6,304	100
I_{b}^4	45	260, 270	12,576	10,928	10.7	49	27	266.26	0.17	376	100
I_{c}	50	340, 350	7,515	8,270	0.867	50	33	344.98	0.019	2	2
I_{d}	40	320, 330	6,135	7,773	8.22	45	23	329.85	8.3	6,733,440	100

Input: A data set $\mathcal{C}$, a set D of descriptors, a set A of different values of λ, and a set $B(\lambda)$ of real numbers close to each $\lambda \in A$;
Output: A subset D^* of D.
for each $\lambda \in A$ **do**
Compute $D_\lambda \coloneqq$ Des-set-LLR$(\mathcal{C}, D, \lambda)$ and $\ell_\lambda \coloneqq \mathrm{R}^2_{\mathrm{CV,MLR}}(\mathcal{C}, D_\lambda, p_1)$
end for;
Set λ_{best} to be a $\lambda \in A$ that maximizes ℓ_λ; Denote $B(\lambda_{\mathrm{best}})$ by $\{\lambda_1, \lambda_2, \ldots, \lambda_{16}\}$;
for each $i \in [1, 16]$ **do**
Compute $D_i \coloneqq$ Des-set-LLR$(\mathcal{C}, D, \lambda_i)$ and let $(w, b), w \in \mathbb{R}^{|D|}, b \in \mathbb{R}$ be the hyperplane obtained by this LLR;
if $|D_i| \leq s_{\max}$ **then**
$D_i' \coloneqq D_i$
else
Let D_i' consist of $s_{\max}$ descriptors $d \in D_i$ that have the $s_{\max}$ largest absolute
values $|w(d)|$ in the weight sets $\{w(d) \mid d \in D_i\}$ of the hyperplane (w, b)
end if;
$D_i^{\dagger} \coloneqq$BS-Reduce$(\mathcal{C}, D_i', p_2)$; $\ell_i \coloneqq \mathrm{R}^2_{\mathrm{CV,MLR}}(\mathcal{C}, D_i^{\dagger}, p_3)$
end for;
Output D^* to be a set $D_i^{\dagger}$ that maximizes $\ell_i, i \in [1, 16]$.

In the computational experiment in this article, we set $A = \{0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 0.01, 0.05, 0.1, 0.5, 0.75, 1, 2, 5, 10, 25, 50, 100\}$, $|B(\lambda)| = 16$ for each $\lambda \in A$, $p_1 \coloneqq p_2 \coloneqq p_3 \coloneqq 5$, and $s_{\max} \coloneqq 150 + 10^4/(|\mathcal{C}| + 200)$.

4 Results and discussion

With our new method of choosing descriptors and formulating an MILP to treat quadratic descriptors in the two-layered model, we implemented the framework for inferring chemical graphs and conducted experiments to evaluate the computational efficiency.

4.1 Chemical properties

In the first phase, we used the following 32 chemical properties of monomers and ten chemical properties of polymers.

For monomers, we used the following data sets: biological half-life (BHL), boiling point (BP), critical temperature (CT), critical pressure (CP), dissociation constants (DC), flash point in closed cup (FP), heat of combustion (HC), heat of vaporization (HV), octanol/water partition coefficient (KOW), melting point (MP), optical rotation (OPTR), refractive index of trees (RFIDT), vapor density (VD) and vapor pressure (VP) provided by HSDB from PubChem (2022); electron density on the most positive atom (EDPA) and Kovats retention index (KOV) provided by Jalali-Heravi and Fatemi (2001); entropy (ET) provided by Duchowicz et al. (2002); heat of atomization (HA) and heat of formation (HF) provided by Roy and Saha (2003); surface tension (SFT) by Goussard et al. (2017); viscosity (VIS) provided by Goussard et al. (2020); isobaric heat capacities liquid (LHCL) and isobaric heat capacities solid (LHCS) provided by Naef (2019); lipophilicity (LP) provided by Xiao (2017); flammable limits lower of organics (FLMLO) provided by Yuan et al. (2019); molar refraction at 20° (MR) provided by Ponce (2003); and solubility (SL) provided by ESOL (MoleculeNet, 2022), and energy of highest occupied molecular orbital (HOMO), energy of lowest unoccupied molecular orbital (LUMO), the energy difference between HOMO and LUMO (GAP), isotropic polarizability (ALPHA), heat capacity at 298.15 K (CV), internal energy at 0 K (U0), and electric dipole moment (MU) provided by ESOL (MoleculeNet, 2022), where the properties from HOMO to MU are based on a common data set QM9.

The data set QM9 contains more than 130,000 compounds. In our experiment, we used a set of 1,000 compounds randomly selected from the data set. For the HV property, we removed the chemical compound with the compound identifier (CID) = 7947 as an extremal outlier from the original data set.

For polymers, we used the following data provided by Bicerano (2002): experimental amorphous density (AMD), characteristic ratio (CHAR), dielectric constant (DIEC), dissipation factor (DISF), heat capacity in liquid (HCL), heat capacity in solid (HCS), mol volume (MLV), permittivity (PRM), refractive index of polymers (RFIDP), and glass transition (TG), where we excluded from our test data set every polymer whose chemical formula could not be found by its name in the book (Bicerano, 2002). A summary of these properties is given in Tables 1, 2. We remark that the previous learning experiments for $\pi \in \{\mathrm{CHAR}, \mathrm{RFIDP}\}$ based on the two-layered model proposed by Azam et al. (2021a) and Zhu et al. (2022c) excluded some number of polymers as outliers. In our experiments, we do not exclude any polymer from the original data set as outliers for these properties.

TABLE 6 Results of inferring a chemical graph $\mathbb{C}^{\dagger}$ and generating recombination solutions for DC with Λ_7.

inst.	n_{LB}	$\underline{y}^*, \bar{y}^*$	#v	#c	I-time	n	n^{int}	η	D-time	$\mathbb{C}$-LB	#$\mathbb{C}$
I_a	30	0.55, 0.60	10,194	9787	3.91	41	25	0.558	0.069	2	2
I_b^1	35	1.10, 1.15	10,415	7,368	4.73	35	11	1.104	0.10	16	16
I_b^2	45	6.00, 6.05	12,976	10,481	57.4	45	25	6.04	0.12	2040	100
I_b^3	45	1.45, 1.50	2,767	10,488	39.7	49	26	1.488	0.28	21,600	100
I_b^4	45	6.10, 6.15	12,558	10,494	26.4	46	25	6.10	0.027	2	2
I_c	50	12.35, 12.40	7,207	7,819	1.75	50	34	12.38	0.020	2	2
I_d	40	3.15, 3.20	5827	7,325	14.9	41	23	3.199	0.079	18,952	100

4.2 Experimental setup and setting data set

We executed the experiments on a PC with a Core i7-9700 (3.0 GHz; 4.7 GHz at the maximum processor and 16 GB RAM DDR4 memory. Prediction functions were constructed using `scikit-learn` version 1.0.2 with Python 3.8.12, and MLPRegressor and rectified linear unit (ReLU) activation function were used for learning based on ANN. The prediction functions constructed by different machine learning models were evaluated using the coefficient of determination R^2. We performed 10 rounds of 5-fold cross-validation, calculated the training and testing R^2 scores for each cross-validation, and recorded the median of the testing R^2 scores across all 50 cross-validations.

For each property π, we first selected a set Λ of chemical elements and then collected a data set $\mathcal{C}_\pi$ on chemical graphs over the set Λ of chemical elements. To construct the data set $\mathcal{C}_\pi$, we eliminated chemical compounds that did not satisfy one of the following: the graph is connected, the number of carbon atoms is at least four, and the number of non-hydrogen neighbors of each atom is at most four.

We set a branch-parameter ρ to be 2, introduce linear descriptors defined by the two-layered graph in the chemical model without suppressing hydrogen, and use the set $D_\pi^{(1)} \cup D_\pi^{(2)}$ of linear and quadratic descriptors (see Zhu et al. (2022a) for the details). We normalize the range of each linear descriptor and the range $\{t \in \mathbb{R} \mid \underline{a} \le t \le \bar{a}\}$ of property values $a(\mathbb{C}), \mathbb{C} \in \mathcal{C}_\pi$ by using the min–max normalization.

Among the above properties, we found that the median of the test coefficient of determination R^2 of the prediction function constructed by LLR (Zhu et al., 2022b) or ALR (Zhu et al., 2022c) exceeds 0.98 for the following nine properties of monomers (resp., three properties of polymers): EDPA, HC, HA, HF, LHCL, LHCS, MR, VD, and U0 (resp., HCL, HCS, and MlV). We excluded the above properties from further analysis because they already achieved excellent predictive performance, and further improvement would not provide additional insights. The remaining 23 chemical properties of monomers and seven chemical properties of polymers were used in the following experiments.

Tables 1, 2 show the size and range of data sets that we prepared for each chemical property to construct a prediction function, where we denote the following:

- π: the name of a chemical property used in the experiment.
- Λ: a set of selected elements used in the data set $\mathcal{C}_\pi$; Λ is one of the following 19 sets:
  Λ_1 ={H,C,O}; Λ_2 ={H,C,O, N}; Λ_3 ={H,C,O, $Si_{(4)}$ }; Λ_4 ={H,C,O, N,$S_{(2)}$,F}; Λ_5 ={H,C,O, N, Cl, Pb}; Λ_6 ={H,C,O, N,$Si_{(4)}$,Cl,Br}; Λ_7 ={H,C,O, N,$S_{(2)}$,$S_{(6)}$,Cl}; Λ_8 ={H,C,O, N,$S_{(2)}$,$S_{(4)}$,$S_{(6)}$,Cl}; Λ_9 ={H, $C_{(2)}$,$C_{(3)}$,$C_{(4)}$,$C_{(5)}$,O, $N_{(1)}$, $N_{(2)}$, $N_{(3)}$, F}; Λ_{10} ={H,C,O, N,$P_{(2)}$,$P_{(5)}$,Cl}; Λ_{11} ={H, C, $O_{(1)}$, $O_{(2)}$, N}; Λ_{12} ={H, C, O, N, Cl}; Λ_{13} ={H, C, O, N, Cl, $S_{(2)}$ }; Λ_{14} ={H, C, $O_{(1)}$, $O_{(2)}$, N, Cl, $Si_{(4)}$, F}; Λ_{15} ={H,C,$O_{(1)}$, $O_{(2)}$, N, $Si_{(4)}$,Cl,F $S_{(2)}$, $S_{(6)}$,Br}; Λ_{16} ={H, C, $O_{(2)}$, N, Cl, $P_{(3)}$, $P_{(5)}$, $S_{(2)}$, $S_{(4)}$, $S_{(6)}$, $Si_{(4)}$, Br, I}, where $a_{(i)}$ for a chemical element a, and an integer $i \ge 1$ means a chemical element a with valence i.
- $|\mathcal{C}_\pi|$: the size of data set $\mathcal{C}_\pi$ over Λ for the property π.
- $\underline{n}$, $\bar{n}$: the minimum and maximum values of the number $n(\mathbb{C})$ of non-hydrogen atoms in the compounds $\mathbb{C}$ in $\mathcal{C}_\pi$.
- $\underline{a}$, $\bar{a}$: the minimum and maximum values of $a(\mathbb{C})$ for π over the compounds $\mathbb{C}$ in $\mathcal{C}_\pi$.
- $|\Gamma|$: the number of different edge-configurations of interior edges over the compounds in $\mathcal{C}_\pi$.
- $|\mathcal{F}|$: the number of non-isomorphic chemical rooted trees in the set of all 2-fringe-trees in the compounds in $\mathcal{C}_\pi$.
- K_1: the size $|D_\pi^{(1)}|$ of a set $D_\pi^{(1)}$ of linear descriptors, where $|D_\pi^{(2)}| = (3(K_1)^2 + K_1)/2$ holds.

4.3 Results on the first phase of the framework

For each chemical property π, we construct a prediction function by one of the following four methods (i)–(iv) and compare their results.

(i) LLR: use lasso linear regression on the set $D_\pi^{(1)}$ of linear descriptors (see Zhu et al. (2022b) for the details of the implementation);
(ii) ANN: use ANN on the set $D_\pi^{(1)}$ of linear descriptors (see Zhu et al. (2022b) for the details of the implementation);
(iii) ALR: use adjustive linear regression on the set $D_\pi^{(1)}$ of linear descriptors (see Zhu et al. (2022c) for the details of the implementation); and
(iv) R-MLR: apply our method (see Zhu et al. (2022a)) of reducing descriptors to the set $D_\pi^{(1)} \cup D_\pi^{(2)}$ of linear and quadratic descriptors, and use multi-linear regression for the resulting set of descriptors.

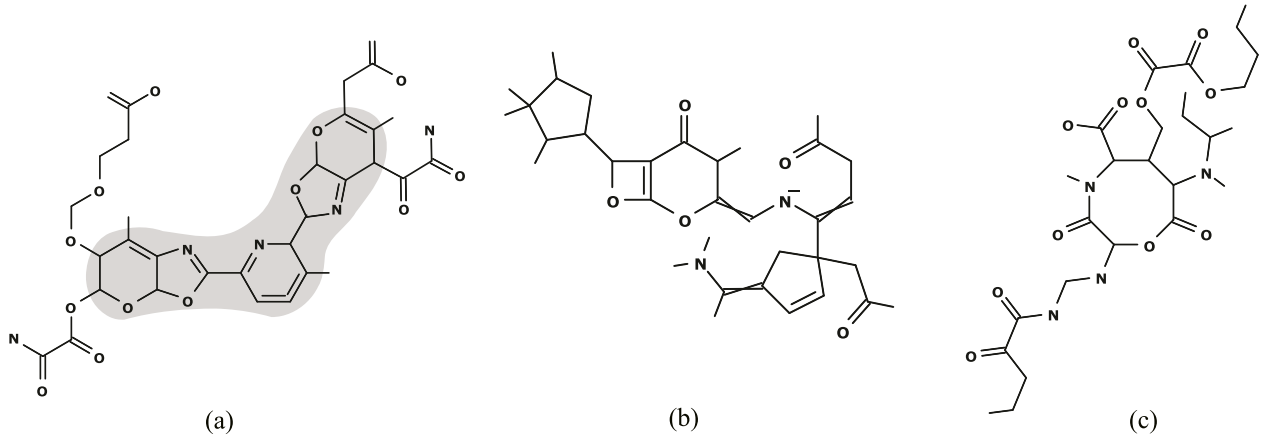

FIGURE 4
(A) $\mathbb{C}^\dagger$ with $\eta(f(\mathbb{C}^\dagger)) = 344.98$ inferred from I_c with $(\underline{y}^*, \bar{y}^*) = (340, 350)$ of BP; **(B)** $\mathbb{C}^\dagger$ with $\eta(f(\mathbb{C}^\dagger)) = 0.558$ inferred from I_a with $(\underline{y}^*, \bar{y}^*) = (0.55, 0.60)$ of DC; and **(C)** $\mathbb{C}^\dagger$ with $\eta(f(\mathbb{C}^\dagger)) = 3.199$ inferred from I_d with $(\underline{y}^*, \bar{y}^*) = (3.15, 3.20)$ of DC.

For methods (i)–(iii), we use the same implementation described in Zhu et al. (2022b) and Zhu et al. (2022c) and omit the details.

In method (iv), we apply LLR-Reduce$(\mathcal{C}_\pi, D_\pi^{(1)} \cup D_\pi^{(2)})$ to compute $\tilde{D} \subset D_\pi^{(1)} \cup D_\pi^{(2)}$ of size 5000 if $|D_\pi^{(1)} \cup D_\pi^{(2)}| > 5000$ for a monomer property π. In the other case, we set $\tilde{D} \coloneqq D_\pi^{(1)} \cup D_\pi^{(2)}$. Then, apply Select-Des-set$(\mathcal{C}_\pi, \tilde{D})$ to get $D^* \subseteq \tilde{D}$, which is used to construct prediction functions based on MLR.

Results of the learning experiments are listed in Tables 3, 4, where:

- π: the property name;
- Λ: the chemical element set of $\mathcal{C}_\pi$;
- LLR, ANN, ALR, R-MLR: the median of test R^2 score in ten times 5-fold cross-validations for functions obtained by methods (i), (ii), (iii), and (iv), respectively;
- the best R^2 score for each property among LLR, ANN, ALR, and R-MLR is indicated by "*";
- K_1^*, K_2^*: the number K_1^* of reduced linear descriptors and the number K_2^* of reduced quadratic descriptors in D^* used in MLR.

The computation times of finding D^* and constructing functions by our method (iv) were in the ranges of $[80, 4 \times 10^4]$ seconds and $[0.03, 0.46]$ seconds, respectively.

Tables 3, 4 show that method (iv) significantly increased the learning performance of several properties and achieved the best scores among methods (i)–(iii) for 43 of 47 data sets. We also noticed that most of the selected descriptors in D^* are quadratics, which confirms the effectiveness of the proposed quadratic descriptors.

4.4 Results on the second phase of the framework

To execute the second phase, we used a set of seven instances I_a, $I_b^i, i \in [1, 4]$, I_c, and I_d based on the seed graphs prepared by Zhu et al. (2022b). We here present their seed graphs G_C (see Zhu et al. (2022a)) for the details of I_a, $I_b^i, i \in [1, 4]$, I_c, and I_d).

The seed graph G_C^1 of I_b^1 (resp., G_C^i, $i = 2, 3, 4$ of I_b^i, $i = 2, 3, 4$) is illustrated in Figure 2.

Instance I_c is introduced to infer an intermediate graph $\mathbb{C}^\dagger$, which preserves

- the core part of $\mathbb{C}_A$: CID 24822711 given in Figure 3A; and
- the frequencies of all edge-configurations of $\mathbb{C}_B$: CID 59170444 given in Figure 3B.The seed graph G_C of this instance is depicted in gray in Figure 3A).

Instance I_d has been introduced in order to infer a chemical graph $\mathbb{C}^\dagger$ such that

- $\mathbb{C}^\dagger$ is monocyclic (where the seed graph of I_d is given by G_C^1 in Figure 2(i)); and
- the frequency vector of edge configurations in $\mathbb{C}^\dagger$ is a vector obtained by merging those of chemical graphs $\mathbb{C}_A$: CID 10076784 and $\mathbb{C}_B$: CID 44340250 in Figures 3C, D, respectively.

4.4.1 Solving an MILP for the inverse problem

We executed the stage of solving an MILP to infer a chemical graph for two properties $\pi \in \{\text{BP, DC}\}$.

For the MILP formulation $\mathcal{M}_{f,\eta,\sigma}$, we use the prediction function η for each $\pi \in \{\text{BP, DC}\}$ by method (iv), R-MLR that attained the median test R^2 in Table 3. To solve an MILP with the formulation, we used CPLEX version 12.10. Tables 5, 6 show the computational results of the experiment in this stage for the two properties, where we denote the following:

- n_{LB}: a lower bound on the number of non-hydrogen atoms in a chemical graph $\mathbb{C}$ to be inferred;
- $\underline{y}^*$, $\bar{y}^*$: lower and upper bounds $\underline{y}^*, \bar{y}^* \in \mathbb{R}$ on the value $a(\mathbb{C})$ of a chemical graph $\mathbb{C}$ to be inferred;
- #v (resp., #c): the number of variables (resp., constraints) in the MILP;
- I-time: the time (sec.) to solve the MILP;
- n: the number $n(\mathbb{C}^\dagger)$ of non-hydrogen atoms in the chemical graph $\mathbb{C}^\dagger$ inferred by solving the MILP;
- n^{int}: the number $n^{\text{int}}(\mathbb{C}^\dagger)$ of interior vertices in the chemical graph $\mathbb{C}^\dagger$; and
- η: the predicted property value $\eta(f(\mathbb{C}^\dagger))$ of the chemical graph $\mathbb{C}^\dagger$.

TABLE 7 Results of generating neighbor solutions of $\mathbb{C}^\dagger$.

(inst., π)	n	δ	#sol	#infs	#ign	#TO
(I_c, BP)	50	0.1	5	1	3	39
(I_a, DC)	30	0.1	40	1	0	7
(I_b^4, DC)	45	0.1	2	0	0	46
(I_c, DC)	40	0.05	0	0	0	48

Figure 4A illustrates the chemical graph $\mathbb{C}^\dagger$ inferred from I_c with $(\underline{y}^\star, \bar{y}^\star) = (340, 350)$ of BP in Table 5.

Figure 4B (resp., Figure 4C) illustrates the chemical graph $\mathbb{C}^\dagger$ inferred from I_a with $(\underline{y}^\star, \bar{y}^\star) = (0.55, 0.60)$ (resp., I_d with $(\underline{y}^\star, \bar{y}^\star) = (3.15, 3.20)$) of DC in Table 6.

In this experiment, we prepared several different types of instances: instances I_a and I_c have restricted seed graphs, the other instances have abstract seed graphs, and instances I_c and I_d have restricted sets of fringe trees. From Tables 5, 6, we observe that an instance with many variables and constraints takes more running time than those with a smaller size in general. All instances in this experiment are solved in a few seconds to approximately 60 s with our MILP formulation.

4.4.2 Generating recombination solutions

Let $\mathbb{C}^\dagger$ be a chemical graph obtained by solving the MILP $\mathcal{M}_{f,\eta,\sigma}$ for the inverse problem. We here execute a stage of generating recombination solutions $\mathbb{C}^\star \in \mathcal{G}_\sigma$ of $\mathbb{C}^\dagger$ such that $f(\mathbb{C}^\star) = x^\star = f(\mathbb{C}^\dagger)$.

We execute an algorithm for generating chemical isomers of $\mathbb{C}^\dagger$ up to 100 when the number of all chemical isomers exceeds 100. For this, we use a dynamic programming algorithm (Zhu et al., 2022b). The algorithm first decomposes $\mathbb{C}^\dagger$ into a set of acyclic chemical graphs. Next, it replaces each acyclic chemical graph T with another acyclic chemical graph T' that admits the same feature vector as that of T, and finally, it assembles the resulting acyclic chemical graphs into a chemical isomer $\mathbb{C}^\star$ of $\mathbb{C}^\dagger$. The algorithm can compute a lower bound on the total number of all chemical isomers $\mathbb{C}^\dagger$ without generating all of them.

Tables 5, 6 show the computational results of the experiment in this stage for the two properties $\pi \in$ {BP, DC}, where we denote the following:

- D-time: the running time (s) to execute the dynamic programming algorithm to compute a lower bound on the number of all chemical isomers $\mathbb{C}^\star$ of $\mathbb{C}^\dagger$ and generate all (or up to 100) chemical isomers $\mathbb{C}^\star$;
- $\mathbb{C}$-LB: a lower bound on the number of all chemical isomers $\mathbb{C}^\star$ of $\mathbb{C}^\dagger$; and
- #$\mathbb{C}$: the number of all (or up to 100) chemical isomers $\mathbb{C}^\star$ of $\mathbb{C}^\dagger$ generated in this stage.

From Tables 5, 6, we observe the running time and the number of generated recombination solutions in this stage.

The chemical graph $\mathbb{C}^\dagger$ in I_b^2, I_b^3, and I_d admits a large number of chemical isomers $\mathbb{C}^\star$ in some cases, where a lower bound $\mathbb{C}$-LB on the number of chemical isomers is derived without generating all of them. For the other instances, the running time for generating up to 100 target chemical graphs in this stage is less than 0.03 s. For some chemical graphs $\mathbb{C}^\dagger$, the number of chemical isomers found by our algorithm was small. This is because some of the acyclic chemical graphs in the decomposition of $\mathbb{C}^\dagger$ have no alternative acyclic chemical graph other than the original one.

4.4.3 Generating neighbor solutions

Let $\mathbb{C}^\dagger$ be a chemical graph obtained by solving the MILP $\mathcal{M}_{f,\eta,\sigma}$ for the inverse problem. We executed a stage of generating neighbor solutions of $\mathbb{C}^\dagger$.

We select an MILP for the inverse problem with a prediction function η such that a solution $\mathbb{C}^\dagger$ of the MILP admits only two isomers $\mathbb{C}^\star$ in the stage of generating recombination solutions; that is, instance I_c for property BP with Λ_7 and instances I_a, I_b^4 and I_c for property DC with Λ_7.

In this experiment, we add to the MILP $\mathcal{M}_{f,\eta,\sigma}$ an additional set Θ of two linear constraints on linear and quadratic descriptors as follows. For the two constraints, we use the prediction functions η_π constructed by R-MLR for properties $\pi \in$ {LP, SL} with Λ_8 in Table 3.

Let $D_\pi^\star$ denote the set of descriptors selected in the construction of prediction function for properties $\pi \in$ {BP, DC} with Λ_7 and $\pi \in$ {LP, SL} with Λ_8 in Table 3, and let D_π^{union}, $\pi \in$ {BP, DC} denote the union $D_\pi^\star \cup D_{\text{LP}}^\star \cup D_{\text{SL}}^\star$.

We regard each of η_{LP} and η_{SL} as a function from $\mathbb{R}^{|D_\pi^{\text{union}}|}$ to $\mathbb{R}$ for $\pi \in$ {BP, DC}. We set $p_{\text{dim}} := 2$ and let Θ consist of two linear constraints $\theta_1 := \eta_{\text{LP}}$ and $\theta_2 := \eta_{\text{SL}}$. We set $\delta := 0.1$ or 0.05, which defines a two-dimensional grid space where $\mathbb{C}^\dagger$ is mapped to the origin (see Azam et al. (2021a) for the details on the neighbors). In these experiments, we check the feasibility of 48 neighbors of the origin $\mathbb{C}^\dagger$ in a grid in an increasing order w.r.t. the distance. The time limit of the solver is set to be 300 s. We do not check the feasibility of a neighbor z and ignore it if there exists a neighbor z' for which the MILP formulation is infeasible and z' is closer to $\mathbb{C}^\dagger$ than z.The results of these experiments are listed in Table 7 where

- (inst., π): specification I and property π;
- n: the number of atoms in the hydrogen-suppressed test instance;
- δ: the size of a sub-region in the grid;
- #sol: the number of new graphs inferred from 48 neighbors;
- #infs: the number of infeasible neighbors;
- #ign: the number of ignored neighbors;
- #TO: the number of neighbors for which the running time of the solver exceeds the time limit.

The branch-and-bound method for solving an MILP sometimes takes an extremely large execution time for the same size of instances. We introduce a time limit to bound an entire running time to skip such instances when testing the feasibility of neighbors

in N_0. From Table 7, we observe that some number of neighbor solutions of the solution $\mathbb{C}^\dagger$ to the MILP $\mathcal{M}_{f,\eta,\sigma}$ could be generated for each of the four instances.

5 Conclusion

In the framework of inferring chemical graphs, the descriptors of a prediction function were mainly defined to be the frequencies of local graph structures in the two-layered model, and such a definition was important to derive a compact MILP formulation for inferring a desired chemical graph. To improve the performance of prediction functions in the framework, this article introduced a multiplication of two of these descriptors as a new descriptor and examined the effectiveness of the new set of descriptors. For this, we designed a method for reducing the size of a descriptor set to not lose the learning performance in constructing prediction functions and gave a compact formulation to compute a product of two values in an MILP. From the results of our computational experiments, we observe that a prediction function constructed by our new method performs considerably better than the prediction functions constructed by the previous methods for several chemical properties. Furthermore, the modified MILP, with the computation of quadratic descriptors, was able to infer desired chemical graphs with up to 50 non-hydrogen atoms.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

Author contributions

JZ: data curation, funding acquisition, software, validation, and writing–review and editing. NA: data curation, formal analysis, software, validation, and writing–review and editing. SC: software, validation, and writing–review and editing. RI: software, validation, and writing–review and editing. KH: data curation, software, and writing–review and editing. LZ: data curation, software, and writing–review and editing. HN: conceptualization, data curation, formal analysis, methodology, project administration, and writing–original draft. TA: conceptualization, data curation, funding acquisition, methodology, and writing–review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. The publication cost of this article is funded by the Japan Society for the Promotion of Science, Japan, under grant numbers 22H00532, 22K19830, and 22KJ1979. The funding agency has no role in the conceptualization, design, data collection, analysis, decision to publish, or preparation of the manuscript.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Publisher's note

References

Akutsu, T., Fukagawa, D., Jansson, J., and Sadakane, K. (2012). Inferring a graph from path frequency. *Discrete Appl. Math.* 160, 1416–1428. doi:10.1016/j.dam.2012.02.002

Azam, N. A., Zhu, J., Haraguchi, K., Zhao, L., Nagamochi, H., and Akutsu, T. (2021a). "Molecular design based on artificial neural networks, integer programming and grid neighbor search," in 2021 IEEE international Conference on Bioinformatics and biomedicine (BIBM) *(IEEE)*, 360–363.

Azam, N. A., Zhu, J., Sun, Y., Shi, Y., Shurbevski, A., Zhao, L., et al. (2021b). A novel method for inference of acyclic chemical compounds with bounded branch-height based on artificial neural networks and integer programming. *Algorithms Mol. Biol.* 16, 18. doi:10.1186/s13015-021-00197-2

Bicerano, J. (2002). *Prediction of polymer properties*. Boca Raton: cRc Press.

Bohacek, R. S., McMartin, C., and Guida, W. C. (1996). The art and practice of structure-based drug design: a molecular modeling perspective. *Med. Res. Rev.* 16, 3–50. doi:10.1002/(SICI)1098-1128(199601)16:1<3::AID-MED1>3.0.CO;2-6

Catacutan, D. B., Alexander, J., Arnold, A., and Stokes, J. M. (2024). Machine learning in preclinical drug discovery. *Nat. Chem. Biol.* 20, 960–973.

Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., et al. (2014). Qsar modeling: where have you been? where are you going to? *J. Med. Chem.* 57, 4977–5010. doi:10.1021/jm4004285

De Cao, N., and Kipf, T. (2018). Molgan: an implicit generative model for small molecular graphs. *arXiv Prepr. arXiv:1805.*

Draper, N. R., and Smith, H. (1998). *Applied regression analysis*, 326. John Wiley and Sons.

Duchowicz, P., Castro, E. A., and Toropov, A. A. (2002). Improved qspr analysis of standard entropy of acyclic and aromatic compounds using optimized correlation weights of linear graph invariants. *Comput. and Chem.* 26, 327–332. doi:10.1016/s0097-8485(01)00121-8

Gartner III, T. E., Ferguson, A. L., and Debenedetti, P. G. (2024). Data-driven molecular design and simulation in modern chemical engineering. *Nat. Chem. Eng.* 1, 6–9.

Ghasemi, F., Mehridehnavi, A., Pérez-Garrido, A., and Pérez-Sánchez, H. (2018). Neural network and deep-learning algorithms used in qsar studies: merits and drawbacks. *Drug Discov. today* 23, 1784–1790. doi:10.1016/j.drudis.2018.06.016

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., et al. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS central Sci.* 4, 268–276. doi:10.1021/acscentsci.7b00572

Goussard, V., Duprat, F., Gerbaud, V., Ploix, J.-L., Dreyfus, G., Nardello-Rataj, V., et al. (2017). Predicting the surface tension of liquids: comparison of four modeling approaches and application to cosmetic oils. *J. Chem. Inf. Model.* 57, 2986–2995. doi:10.1021/acs.jcim.7b00512

Goussard, V., Duprat, F., Ploix, J.-L., Dreyfus, G., Nardello-Rataj, V., and Aubry, J.-M. (2020). A new machine-learning tool for fast estimation of liquid viscosity. application to cosmetic oils. *J. Chem. Inf. Model.* 60, 2012–2023. doi:10.1021/acs.jcim.0c00083

Ido, R., Cao, S., Zhu, J., Azam, N. A., Haraguchi, K., Zhao, L., et al. (2021). A method for inferring polymers based on linear regression and integer programming. *arXiv preprint arXiv:2109.02628*.

Ikebata, H., Hongo, K., Isomura, T., Maezono, R., and Yoshida, R. (2017). Bayesian molecular design with a chemical language model. *J. computer-aided Mol. Des.* 31, 379–391. doi:10.1007/s10822-016-0008-z

Jalali-Heravi, M., and Fatemi, M. H. (2001). Artificial neural network modeling of kovats retention indices for noncyclic and monocyclic terpenes. *J. Chromatogr. A* 915, 177–183. doi:10.1016/s0021-9673(00)01274-7

Kim, J., Park, S., Min, D., and Kim, W. (2021). Comprehensive survey of recent drug discovery using deep learning. *Int. J. Mol. Sci.* 22, 9983. doi:10.3390/ijms22189983

Kipf, T. N., and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv Prepr. arXiv:1609.02907*.

Kusner, M. J., Paige, B., and Hernández-Lobato, J. M. (2017). "Grammar variational autoencoder," in *International conference on machine learning* (Sydney, Australia: PMLR), 1945–1954.

Lo, Y.-C., Rensi, S. E., Torng, W., and Altman, R. B. (2018). Machine learning in chemoinformatics and drug discovery. *Drug Discov. today* 23, 1538–1546. doi:10.1016/j.drudis.2018.05.010

Madhawa, K., Ishiguro, K., Nakago, K., and Abe, M. (2019). Graphnvp: an invertible flow model for generating molecular graphs. *arXiv preprint arXiv:1905.11600*.

Miyao, T., and Funatsu, K. (2024). Data-driven molecular structure generation for inverse qspr/qsar problem. *In Drug Development Supported by Informatics (Springer)*. 47–59.

Miyao, T., Kaneko, H., and Funatsu, K. (2016). Inverse qspr/qsar analysis for chemical structure generation (from y to x). *J. Chem. Inf. Model.* 56, 286–299. doi:10.1021/acs.jcim.5b00628

MoleculeNet (2022). Available at: https://moleculenet.org/datasets-1.

Naef, R. (2019). Calculation of the isobaric heat capacities of the liquid and solid phase of organic compounds at and around 298.15 k based on their "true" molecular volume. *Molecules* 24, 1626. doi:10.3390/molecules24081626

Ponce, Y. M. (2003). Total and local quadratic indices of the molecular pseudograph's atom adjacency matrix: applications to the prediction of physical properties of organic compounds. *Molecules* 8, 687–726. doi:10.3390/80900687

PubChem (2022). Annotations from hsdb. Available at: https://pubchem.ncbi.nlm.nih.gov/.

Roy, K., and Saha, A. (2003). Comparative qspr studies with molecular connectivity, molecular negentropy and tau indices: Part i: molecular thermochemical properties of diverse functional acyclic compounds. *J. Mol. Model.* 9, 259–270. doi:10.1007/s00894-003-0135-z

Rupakheti, C., Virshup, A., Yang, W., and Beratan, D. N. (2015). Strategy to discover diverse optimal molecules in the small molecule universe. *J. Chem. Inf. Model.* 55, 529–537. doi:10.1021/ci500749q

Segler, M. H., Kogej, T., Tyrchan, C., and Waller, M. P. (2018). Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central Sci.* 4, 120–131. doi:10.1021/acscentsci.7b00512

Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., and Tang, J. (2020). Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv Prepr. arXiv:2001*, 09382.

Shi, Y., Zhu, J., Azam, N. A., Haraguchi, K., Zhao, L., Nagamochi, H., et al. (2021). An inverse qsar method based on a two-layered model and integer programming. *Int. J. Mol. Sci.* 22, 2847. doi:10.3390/ijms22062847

Tetko, I. V., and Engkvist, O. (2020). From big data to artificial intelligence: chemoinformatics meets new challenges. *J. Cheminformatics* 12, 1–3. doi:10.1186/s13321-020-00475-y

Xiao, N. (2017). *Lipophilicity dataset - logd7*, 4 of 1. 130 compounds.

Xiong, J., Xiong, Z., Chen, K., Jiang, H., and Zheng, M. (2021). Graph neural networks for automated *de novo* drug design. *Drug Discov. today* 26, 1382–1393. doi:10.1016/j.drudis.2021.02.011

Yang, X., Zhang, J., Yoshizoe, K., Terayama, K., and Tsuda, K. (2017). Chemts: an efficient python library for *de novo* molecular generation. *Sci. Technol. Adv. Mater.* 18, 972–976. doi:10.1080/14686996.2017.1401424

Yuan, S., Jiao, Z., Quddus, N., Kwon, J. S.-I., and Mashuga, C. V. (2019). Developing quantitative structure–property relationship models to predict the upper flammability limit using machine learning. *Industrial and Eng. Chem. Res.* 58, 3531–3537. doi:10.1021/acs.iecr.8b05938

Zhu, J., Azam, N. A., Cao, S., Ido, R., Haraguchi, K., Zhao, L., et al. (2022a). Molecular design based on integer programming and quadratic descriptors in a two-layered model. *arXiv Prepr. arXiv:2209.13527*.

Zhu, J., Azam, N. A., Haraguchi, K., Zhao, L., Nagamochi, H., and Akutsu, T. (2022b). "A method for molecular design based on linear regression and integer programming," in *Proceedings of the 2022 12th international conference on bioscience, biochemistry and bioinformatics*, 21–28.

Zhu, J., Haraguchi, K., Nagamochi, H., and Akutsu, T. (2022c). Adjustive linear regression and its application to the inverse qsar. *BIOINFORMATICS*, 144–151. doi:10.5220/0010853700003123

Glossary

Alpha	isotropic polarizability
ALR	adjustive linear regression
AmD	experimental amorphous density
ANN	artificial neural network
BHL	biological half life
Bp	boiling point
ChaR	characteristic ratio
Cp	critical pressure
Ct	critical temperature
Cv	heat capacity at 298.15 K
Dc	dissociation constants
DieC	dielectric constant
DisF	dissipation factor
EDPA	electron density on the most positive atom
ET	entropy
FlmLO	flammable limits lower of organics
Fp	flash point in closed cup
Gap	the energy difference between HOMO and LUMO
Ha	heat of atomization
Hc	heat of combustion
HcL	heat capacity in liquid
HcS	heat capacity in solid
Hf	heat of formation
HOMO	energy of highest-occupied molecular orbital
Hv	heat of vaporization
Kov	Kovats retention index
Kow	octanol/water partition coefficient
LLR	lasso linear regression
Lp	lipophilicity
LUMO	energy of lowest-unoccupied molecular orbital
MILP	mixed-integer linear programming
MLR	multidimensional linear regression
MlV	mol volume
Mp	melting point
Mr	molar refraction at 20° degree
mu	electric dipole moment
OptR	optical rotation
Prm	permittivity
QSAR	quantitative structure–activity relationship
RfIdP	refractive index of polymers
RfIdT	refractive index of trees
SfT	surface tension
Sl	solubility
Tg	glass transition
U0	internal energy at 0 K
Vd	vapor density
Vis	viscosity
Vp	vapor pressure