Frontiers in Bioscience Landmark

Front. Biosci. (Landmark Ed) **2022**; 27(6): 188
https://doi.org/10.31083/j.fbl2706188

Original Research

An Inverse QSAR Method Based on Linear Regression and Integer Programming

Jianshen Zhu[1,*,†], Naveed Ahmed Azam[1,†], Kazuya Haraguchi[1,†], Liang Zhao[2], Hiroshi Nagamochi[1,†], Tatsuya Akutsu[3]

[1]Department of Applied Mathematics and Physics, Kyoto University, 606-8501 Kyoto, Japan
[2]Graduate School of Advanced Integrated Studies in Human Survavibility (Shishu-Kan), Kyoto University, 606-8306 Kyoto, Japan
[3]Bioinformatics Center, Institute for Chemical Research, Kyoto University, 611-0011 Uji, Japan
*Correspondence: zhujs@amp.i.kyoto-u.ac.jp (Jianshen Zhu)
† These authors contributed equally.

Academic Editors: Agnieszka Kaczor and Graham Pawelec
Submitted: 16 February 2022 Revised: 28 March 2022 Accepted: 7 April 2022 Published: 10 June 2022

Abstract

Background: Drug design is one of the important applications of biological science. Extensive studies have been done on computer-aided drug design based on inverse quantitative structure activity relationship (inverse QSAR), which is to infer chemical compounds from given chemical activities and constraints. However, exact or optimal solutions are not guaranteed in most of the existing methods. **Method**: Recently a novel framework based on artificial neural networks (ANNs) and mixed integer linear programming (MILP) has been proposed for designing chemical structures. This framework consists of two phases: an ANN is used to construct a prediction function, and then an MILP formulated on the trained ANN and a graph search algorithm are used to infer desired chemical structures. In this paper, we use linear regression instead of ANNs to construct a prediction function. For this, we derive a novel MILP formulation that simulates the computation process of a prediction function by linear regression. **Results**: For the first phase, we performed computational experiments using 18 chemical properties, and the proposed method achieved good prediction accuracy for a relatively large number of properties, in comparison with ANNs in our previous work. For the second phase, we performed computational experiments on five chemical properties, and the method could infer chemical structures with around up to 50 non-hydrogen atoms. **Conclusions**: Combination of linear regression and integer programming is a potentially useful approach to computational molecular design.

Keywords: machine learning; linear regression; integer programming; chemoinformatics; materials informatics; QSAR/QSPR; molecular design

1. Introduction

Analysis of the activities and properties of chemical compounds is important not only for chemical science but also for biological science because chemical compounds play important roles in metabolic and many other pathways. Computational prediction of chemical activities from their structural data has been studied for several decades under the name of *quantitative structure activity relationship* (QSAR) [1]. In addition to traditional regression-based methods, various machine learning methods have been applied to QSAR [2,3]. Recently, neural networks and deep-learning technologies have extensively been applied to QSAR [4].

Inference of chemical structures with desired chemical activities under some constraints is also important because of its potential applications to drug design, and the problem has been studied under the name of *inverse quantitative structure activity relationship* (inverse QSAR). Chemical compounds are commonly represented by undirected graphs called *chemical graphs* in which vertices and edges correspond to atoms and chemical bonds, respectively. Due to the difficulty of directly handling chemical graphs in both QSAR and inverse QSAR, chemical compounds are usually represented as vectors of integer or real numbers, which are called *descriptors* in chemoinformatics and correspond to *feature vectors* in machine learning. In inverse QSAR, one major approach is to first infer feature vectors from given chemical activities and constraints, and then reconstruct chemical structures from these feature vectors [5–7]. However, the reconstruction itself is not an easy task because the number of possible chemical graphs is huge. For example, the number of chemical graphs with up to 30 atoms (vertices) C, N, O, and S may exceed 10^{60} [8]. Indeed, the problem to infer a chemical graph from a given feature vector is known as a computationally difficult problem (precisely, NP-hard) except for some simple cases [9]. Most existing methods for inverse QSAR do not guarantee exact or optimal solutions due to these inherent difficulties.

Recently, *artificial neural networks* (ANNs), in particular, graph convolutional networks [10] are extensively used for inverse QSAR. For example, recurrent neural networks [11,12], variational autoencoders [13], grammar variational autoencoders [14], invertible flow models [15,16], and generative adversarial networks [17] have been applied. However, these methods do not yet guarantee exact or optimal solutions.

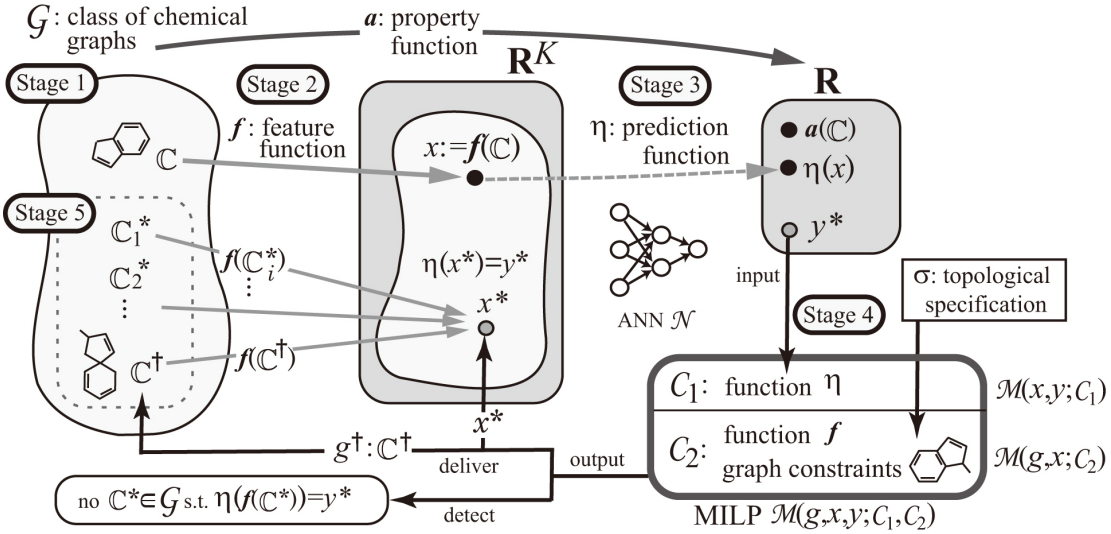


Fig. 1. An illustration of a framework for inferring a set of chemical graphs $\mathbb{C}^*$.

Akutsu and Nagamochi [18] proved that the computation process of a given ANN can be simulated as a mixed integer linear programming (MILP). Based on this result, a novel framework for inferring a set of chemical graphs has been developed [19,20], which is illustrated in Fig. 1. This framework consists of two phases. In the first phase, it constructs a prediction function and in the second phase, it infers a chemical graph. There are three stages in the first phase of the framework. In Stage 1, a chemical property π and a class $\mathcal{G}$ of graphs are selected, and a property function a is defined so that $a(\mathbb{C})$ is the value of π for a compound $\mathbb{C} \in \mathcal{G}$. Then we collect a data set D_π of chemical graphs in $\mathcal{G}$ such that $a(\mathbb{C})$ is available for every $\mathbb{C} \in D_\pi$. In Stage 2, a feature function $f : \mathcal{G} \to \mathbb{R}^K$ for a positive integer K is introduced. In Stage 3, a prediction function η is constructed with an ANN $\mathcal{N}$ that, given a vector $x \in \mathbb{R}^K$, returns a value $y = \eta(x) \in \mathbb{R}$ so that $\eta(f(\mathbb{C}))$ serves as a predicted value to $a(\mathbb{C})$ of π for each $\mathbb{C} \in D_\pi$. Given a target chemical value y^*, the second phase consists the next two phases to infer chemical graphs $\mathbb{C}^*$ with $\eta(f(\mathbb{C}^*)) = y^*$. A feature function f and a prediction function η are obtained in the first phase, and we call an additional constraint on the substructures of target chemical graphs a *topological specification*. In Stage 4, the following two MILP formulations are designed:

- MILP $\mathcal{M}(x, y; \mathcal{C}_1)$ with a set $\mathcal{C}_1$ of linear constraints on variables x and y (and some other auxiliary variables) simulates the process of computing $y := \eta(x)$ from a vector x; and
- MILP $\mathcal{M}(g, x; \mathcal{C}_2)$ with a set $\mathcal{C}_2$ of linear constraints on variable x and a variable vector g that represents a chemical graph $\mathbb{C}$ (and some other auxiliary variables) simulates the process of computing $x := f(\mathbb{C})$ from a chemical graph $\mathbb{C}$ and chooses a chemical graph $\mathbb{C}$ that satisfies the given topological specification σ.

Given a target value $y^* \in \mathbb{R}$, the combined MILP $\mathcal{M}(g, x, y; \mathcal{C}_1, \mathcal{C}_2)$ is solved to find a feature vector $x^* \in \mathbb{R}^K$ and a chemical graph $\mathbb{C}^\dagger$ that satisfies the specification σ such that $f(\mathbb{C}^\dagger) = x^*$ and $\eta(x^*) = y^*$ (where if the MILP is infeasible then this suggests that such a desired chemical graph does not exist). In Stage 5, by using the inferred chemical graph $\mathbb{C}^\dagger$, we generate other chemical graphs $\mathbb{C}^*$ such that $\eta(f(\mathbb{C}^*)) = y^*$.

Stage 4 MILP formulations to infer chemical graphs with cycle index 0, 1 and 2 are proposed in [20–23], respectively, but no sophisticated topological specification was available yet. A restricted class of acyclic graphs that is characterized by an integer ρ, called a "branch-parameter" is introduced by Azam *et al.* [21]. This restricted class still covers most of the acyclic chemical compounds in the database. Akutsu and Nagamochi [24] extended the idea to define a restricted class of cyclic graphs, called "ρ-lean cyclic graphs" and introduced a set of flexible rules for describing a topological specification. Tanaka *et al.* [25] used a decision tree instead of ANNs to construct a prediction function η in Stage 3 in the framework and an MILP $\mathcal{M}(x, y; \mathcal{C}_1)$ that simulates the computation process of a decision tree.

Recently Shi *et al.* [26] proposed a new model to deal with an arbitrary graph in the framework called a *two-layered model* to represent the feature of a chemical graph. Also, the set of rules for describing a topological specification in [27] was refined so that a prescribed structure can be included in both of the acyclic and cyclic parts of a chemical graph $\mathbb{C}$. In this model, a chemical graph $\mathbb{C}$ with an integer $\rho \geq 1$, we consider two parts, namely, the exterior and the interior of the hydrogen-suppressed graph $\langle\mathbb{C}\rangle$ that is obtained by removing hydrogen from $\mathbb{C}$. The exterior consists of maximal acyclic induced subgraphs with height at most ρ in $\langle\mathbb{C}\rangle$ and the interior is the connected subgraph

of $\langle\mathbb{C}\rangle$ obtained by excluding the exterior. Shi *et al*. [26] also defined a feature vector $f(\mathbb{C})$ of a chemical graph $\mathbb{C}$ as a combination of the frequency of adjacent atom pairs in the interior and the frequency of chemical acyclic graphs among the set of chemical rooted trees T_u rooted at interior-vertices u. Recently, Tanaka *et al*. [25] extended the model in order to directly treat a chemical graph with hydrogens so that the feature of the exterior can be represented with more variety of chemical rooted trees.

The contribution of this paper is as follows. Firstly, we make a slight modification to a model of chemical graphs proposed by Tanaka *et al*. [25] so that we can treat a chemical element with multi-valence such as sulfur S and a chemical graph with cations and anions. Then, we consider the prediction function. One of the most important factors in the framework is the quality of a prediction function η constructed in Stage 3. Also, overfitting is pointed out as a major issue in ANN-based approaches for QSAR because many parameters need to be optimized for ANNs [4]. In this paper, to construct a prediction function in Stage 3, we use linear regression instead of ANNs or decision trees. A learning algorithm for an ANN may not find a set of weights and biases that minimizes the error function since the algorithm simply iterates modification of the current weights and biases until it terminates at a local minimum value, and linear regression is much simpler than ANNs and decision trees and thereby we regard the performance of a prediction function by linear regression as the basis for other more sophisticated machine learning methods. In this paper, we derive an MILP formulation $\mathcal{M}(x, y; \mathcal{C}_1)$ in Stage 4 to simulate the computation process of a prediction function by linear regression. For an MILP formulation $\mathcal{M}(g, x; \mathcal{C}_2)$ that represents a feature function f and a specification σ in Stage 4, we can use the same formulation proposed by Tanaka *et al*. [25] with a slight modification (the detail of the MILP $\mathcal{M}(g, x; \mathcal{C}_2)$ can be found in **Supplementary Material**). In Stage 5, we can also use the dynamic programming algorithm due to Tanaka *et al*. [25] with a slight modification to generate target chemical graphs $\mathbb{C}^*$ and the details are omitted in this paper.

We implemented the framework based on the refined two-layered model and a prediction function by linear regression. The results of our computational experiments reveal a set of chemical properties to which a prediction function constructed by linear regression on our feature function performs well, in comparison with ANNs in our previous work. We also observe that chemical graphs with up to 50 non-hydrogen atoms can be inferred by the proposed method.

The paper is organized as follows. Section 2 introduces some notions and terminologies on graphs, modeling of chemical compounds and our choice of descriptors. Section 3 describes our modification to the two-layered model. Section 4 reviews the idea of linear regression and formulates an MILP $\mathcal{M}(x, y; \mathcal{C}_1)$ that simulates the computing process of a prediction function constructed by linear regression. Section 5 reports the results of some computational experiments conducted for 18 chemical properties such as vapor density and optical rotation. Section 6 gives conclusions with future work. Some technical details are given in **Supplementary Material**.

2. Preliminary

In this section, we review some notions and terminologies related to graphs, modeling of chemical compounds introduced by Tanaka *et al*. [25] and our choice of descriptors.

Let $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$ denote the sets of reals, non-negative reals, integers and non-negative integers, respectively. For two integers a and b, let $[a, b]$ denote the set of integers i with $a \leq i \leq b$.

Graph Given a graph G, let $V(G)$ and $E(G)$ denote the sets of vertices and edges, respectively. For a subset $V' \subseteq V(G)$ (resp., $E' \subseteq E(G)$) of a graph G, let $G - V'$ (resp., $G - E'$) denote the graph obtained from G by removing the vertices in V' (resp., the edges in E'), where we remove all edges incident to a vertex in V' in $G - V'$. An edge subset $E' \subseteq E(G)$ in a connected graph G is called *separating* (resp., *non-separating*) if $G - E'$ becomes disconnected (resp., $G - E'$ remains connected). The *rank* $\mathrm{r}(G)$ of a connected graph G is defined to be the minimum $|F|$ of an edge subset $F \subseteq E(G)$ such that $G - F$ contains no cycle, where $\mathrm{r}(G) = |E(G)| - |V(G)| + 1$. Observe that $\mathrm{r}(G - E') = \mathrm{r}(G) - |E'|$ holds for any non-separating edge subset $E' \subseteq E(G)$. An edge $e = u_1u_2 \in E(G)$ in a connected graph G is called a *bridge* if $\{e\}$ is separating, i.e., $G - e$ consists of two connected graphs G_i containing vertex u_i, $i = 1, 2$. For a connected cyclic graph G, an edge e is called a *core-edge* if it is in a cycle of G or is a bridge $e = u_1u_2$ such that each of the connected graphs G_i, $i = 1, 2$, of $G - e$ contains a cycle. A vertex incident to a core-edge is called a *core-vertex* of G. A path with two end-vertices u and v is called a u, v-*path*.

A vertex designated in a graph G is called a *root*. In this paper, we designate at most two vertices as roots, and denote by $\mathrm{Rt}(G)$ the set of roots of G. We call a graph G *rooted* (resp., *bi-rooted*) if $|\mathrm{Rt}(G)| = 1$ (resp., $|\mathrm{Rt}(G)| = 2$), where we call G *unrooted* if $\mathrm{Rt}(G) = \emptyset$.

For a graph G possibly with roots, a *leaf-vertex* is defined to be a non-root vertex $v \in V(G) \setminus \mathrm{Rt}(G)$ with degree 1. Call the edge uv incident to a leaf vertex v a *leaf-edge*, and denote by $V_{\mathrm{leaf}}(G)$ and $E_{\mathrm{leaf}}(G)$ the sets of leaf-vertices and leaf-edges in G, respectively. For a graph or a rooted graph G, we define graphs $G_i, i \in \mathbb{Z}_+$ obtained from G by removing the set of leaf-vertices i times so that

$$G_0 := G; \quad G_{i+1} := G_i - V_{\mathrm{leaf}}(G_i), \tag{1}$$

where we call a vertex $v \in V_{\mathrm{leaf}}(G_k)$ a *leaf k-branch* and we say that a vertex $v \in V_{\mathrm{leaf}}(G_k)$ has height $\mathrm{ht}(v) = k$

in G. The height $\mathrm{ht}(T)$ of a rooted tree T is defined to be the maximum of $\mathrm{ht}(v)$ of a vertex $v \in V(T)$. For an integer $k \geq 0$, we call a rooted tree T *k-lean* if T has at most one leaf k-branch. For an unrooted cyclic graph G, we regard that the set of non-core-edges in G induces a collection $\mathcal{T}$ of trees each of which is rooted at a core-vertex, where we call G *k-lean* if each of the rooted trees in $\mathcal{T}$ is k-lean.

Modeling of Chemical Compounds

We introduce a set of chemical elements such as `H` (hydrogen), `C` (carbon), `O` (oxygen), `N` (nitrogen) and so on to represent a chemical compound. To distinguish a chemical element a with multiple valences such as `S` (sulfur), we denote a chemical element a with a valence i by $\mathrm{a}_{(i)}$, where we do not use such a suffix (i) for a chemical element a with a unique valence. Let Λ be a set of chemical elements $\mathrm{a}_{(i)}$. For example, $\Lambda = \{\mathrm{H}, \mathrm{C}, \mathrm{O}, \mathrm{N}, \mathrm{P}, \mathrm{S}_{(2)}, \mathrm{S}_{(4)}, \mathrm{S}_{(6)}\}$. Let $\mathrm{val} : \Lambda \to [1, 6]$ be a valence function. For example, $\mathrm{val}(\mathrm{H}) = 1$, $\mathrm{val}(\mathrm{C}) = 4$, $\mathrm{val}(\mathrm{O}) = 2$, $\mathrm{val}(\mathrm{P}) = 5$, $\mathrm{val}\left(\mathrm{S}_{(2)}\right) = 2$, $\mathrm{val}\left(\mathrm{S}_{(4)}\right) = 4$ and $\mathrm{val}\left(\mathrm{S}_{(6)}\right) = 6$. For each chemical element $\mathrm{a} \in \Lambda$, let $\mathrm{mass}(\mathrm{a})$ denote the mass of a.

To represent a chemical compound, we use a *chemical graph* introduced by Tanaka *et al.* [25], which is defined to be a tuple $\mathbb{C} = (H, \alpha, \beta)$ of a simple, connected undirected graph H and functions $\alpha : V(H) \to \Lambda$ and $\beta : E(H) \to [1, 3]$. The set of atoms and the set of bonds in the compound are represented by the vertex set $V(H)$ and the edge set $E(H)$, respectively. The chemical element assigned to a vertex $v \in V(H)$ is represented by $\alpha(v)$ and the bond-multiplicity between two adjacent vertices $u, v \in V(H)$ is represented by $\beta(e)$ of the edge $e = uv \in E(H)$. We say that two tuples $(H_i, \alpha_i, \beta_i), i = 1, 2$ are *isomorphic* if they admit an isomorphism ϕ, i.e., a bijection $\phi : V(H_1) \to V(H_2)$ such that $uv \in E(H_1), \alpha_1(u) = \mathrm{a}, \alpha_1(v) = \mathrm{b}, \beta_1(uv) = m \leftrightarrow \phi(u)\phi(v) \in E(H_2), \alpha_2(\phi(u)) = \mathrm{a}, \alpha_2(\phi(v)) = \mathrm{b}, \beta_2(\phi(u)\phi(v)) = m$. When H_i is rooted at a vertex $r_i, i = 1, 2$, $(H_i, \alpha_i, \beta_i), i = 1, 2$ are *rooted-isomorphic* (r-isomorphic) if they admit an isomorphism ϕ such that $\phi(r_1) = r_2$.

For a notational convenience, we use a function $\beta_{\mathbb{C}} : V(H) \to [0, 12]$ for a chemical graph $\mathbb{C} = (H, \alpha, \beta)$ such that $\beta_{\mathbb{C}}(u)$ means the sum of bond-multiplicities of edges incident to a vertex u; i.e.,

$$\beta_{\mathbb{C}}(u) \triangleq \sum_{uv \in E(H)} \beta(uv) \tag{2}$$

for each vertex $u \in V(H)$. For each vertex $u \in V(H)$, define the *electron-degree* $\mathrm{eledeg}_{\mathbb{C}}(u)$ to be

$$\mathrm{eledeg}_{\mathbb{C}}(u) \triangleq \beta_{\mathbb{C}}(u) - \mathrm{val}(\alpha(u)). \tag{3}$$

For each vertex $u \in V(H)$, let $\deg_{\mathbb{C}}(u)$ denote the number of vertices adjacent to the vertex u in $\mathbb{C}$.

For a chemical graph $\mathbb{C} = (H, \alpha, \beta)$, let $V_{\mathrm{a}}(\mathbb{C}), \mathrm{a} \in \Lambda$ denote the set vertices $v \in V(H)$ such that $\alpha(v) = \mathrm{a}$ in $\mathbb{C}$ and define the *hydrogen-suppressed chemical graph* $\langle \mathbb{C} \rangle$ to be the graph obtained from H by removing all the vertices $v \in V_{\mathrm{H}}(\mathbb{C})$.

3. Two-layered Model

This section reviews the idea of the two-layered model introduced by Shi *et al.* [26], and describes our modifications to the model.

Let $\mathbb{C} = (H, \alpha, \beta)$ be a chemical graph and $\rho \geq 1$ be an integer, which is called a *branch-parameter*.

A *two-layered model* of $\mathbb{C}$ introduced by Shi *et al.* [26] is a partition of the hydrogen-suppressed chemical graph $\langle \mathbb{C} \rangle$ into an "interior" and an "exterior" in the following way. We call a vertex $v \in V(\langle \mathbb{C} \rangle)$ (resp., an edge $e \in E(\langle \mathbb{C} \rangle)$) of G an *exterior-vertex* (resp., *exterior-edge*) if $\mathrm{ht}(v) < \rho$ (resp., e is incident to an exterior-vertex) and denote the sets of exterior-vertices and exterior-edges by $V^{\mathrm{ex}}(\mathbb{C})$ and $E^{\mathrm{ex}}(\mathbb{C})$, respectively and denote $V^{\mathrm{int}}(\mathbb{C}) = V(\langle \mathbb{C} \rangle) \setminus V^{\mathrm{ex}}(\mathbb{C})$ and $E^{\mathrm{int}}(\mathbb{C}) = E(\langle \mathbb{C} \rangle) \setminus E^{\mathrm{ex}}(\mathbb{C})$, respectively. We call a vertex in $V^{\mathrm{int}}(\mathbb{C})$ (resp., an edge in $E^{\mathrm{int}}(\mathbb{C})$) an *interior-vertex* (resp., *interior-edge*). The set $E^{\mathrm{ex}}(\mathbb{C})$ of exterior-edges forms a collection of connected graphs each of which is regarded as a rooted tree T rooted at the vertex $v \in V(T)$ with the maximum $\mathrm{ht}(v)$. Let $\mathcal{T}^{\mathrm{ex}}(\langle \mathbb{C} \rangle)$ denote the set of these chemical rooted trees in $\langle \mathbb{C} \rangle$. The *interior* $\mathbb{C}^{\mathrm{int}}$ of $\mathbb{C}$ is defined to be the subgraph $(V^{\mathrm{int}}(\mathbb{C}), E^{\mathrm{int}}(\mathbb{C}))$ of $\langle \mathbb{C} \rangle$.

Fig. 2 illustrates an example of a hydrogen-suppressed chemical graph $\langle \mathbb{C} \rangle$. For a branch-parameter $\rho = 2$, the interior of the chemical graph $\langle \mathbb{C} \rangle$ in Fig. 2 is obtained by removing the set of vertices with degree 1 $\rho = 2$ times; i.e., first remove the set $V_1 = \{w_1, w_2, \ldots, w_{14}\}$ of vertices of degree 1 in $\langle \mathbb{C} \rangle$ and then remove the set $V_2 = \{w_{15}, w_{16}, \ldots, w_{19}\}$ of vertices of degree 1 in $\langle \mathbb{C} \rangle - V_1$, where the removed vertices become the exterior-vertices of $\langle \mathbb{C} \rangle$.

For each interior-vertex $u \in V^{\mathrm{int}}(\mathbb{C})$, let $T_u \in \mathcal{T}^{\mathrm{ex}}(\langle \mathbb{C} \rangle)$ denote the chemical tree rooted at u (where possibly T_u consists of vertex u) and define the *ρ-fringe-tree* $\mathbb{C}[u]$ to be the chemical rooted tree obtained from T_u by putting back the hydrogens originally attached to T_u in $\mathbb{C}$. Let $\mathcal{T}(\mathbb{C})$ denote the set of ρ-fringe-trees $\mathbb{C}[u], u \in V^{\mathrm{int}}(\mathbb{C})$. Fig. 3 illustrates the set $\mathcal{T}(\mathbb{C}) = \{\mathbb{C}[u_i] \mid i \in [1, 28]\}$ of the 2-fringe-trees of the example $\mathbb{C}$ in Fig. 2.

Feature Function We extend the feature function of a chemical graph $\mathbb{C}$ introduced by Tanaka *et al.* [25]. The feature of an interior-edge $e = uv \in E^{\mathrm{int}}(\mathbb{C})$ such that $\alpha(u) = \mathrm{a}$, $\deg_{\langle \mathbb{C} \rangle}(u) = d$, $\alpha(v) = \mathrm{b}$, $\deg_{\langle \mathbb{C} \rangle}(v) = d'$ and $\beta(e) = m$ is represented by a tuple $(\mathrm{a}d, \mathrm{b}d', m)$, which is called the *edge-configuration* of the edge e, where we call the tuple $(\mathrm{a}, \mathrm{b}, m)$ the *adjacency-configuration* of the edge e.

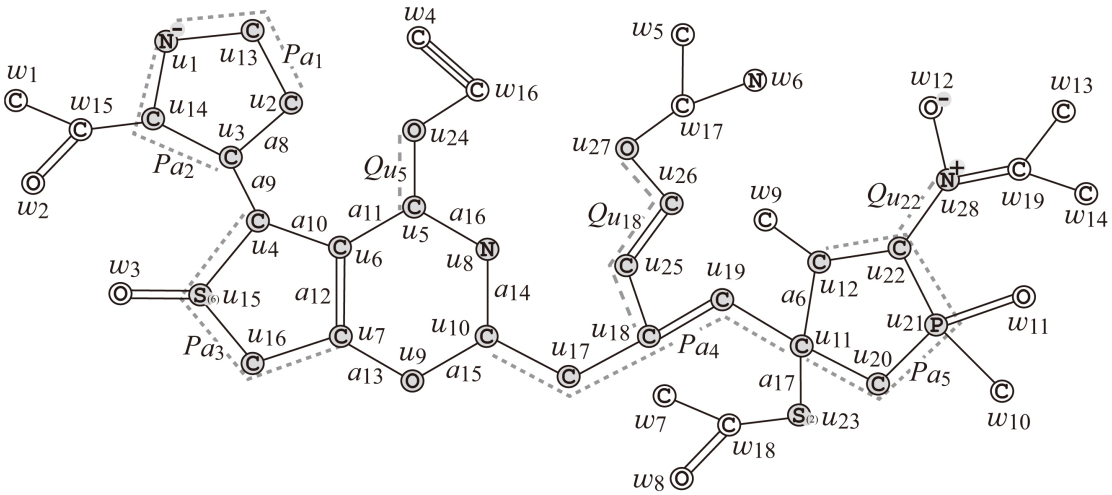


Fig. 2. An illustration of a hydrogen-suppressed chemical graph $\langle\mathbb{C}\rangle$ obtained from a chemical graph $\mathbb{C}$ with $\mathrm{r}(\mathbb{C}) = 4$ by removing all the hydrogens, where for $\rho = 2$, $V^{\mathrm{ex}}(\mathbb{C}) = \{w_i \mid i \in [1, 19]\}$ and $V^{\mathrm{int}}(\mathbb{C}) = \{u_i \mid i \in [1, 28]\}$

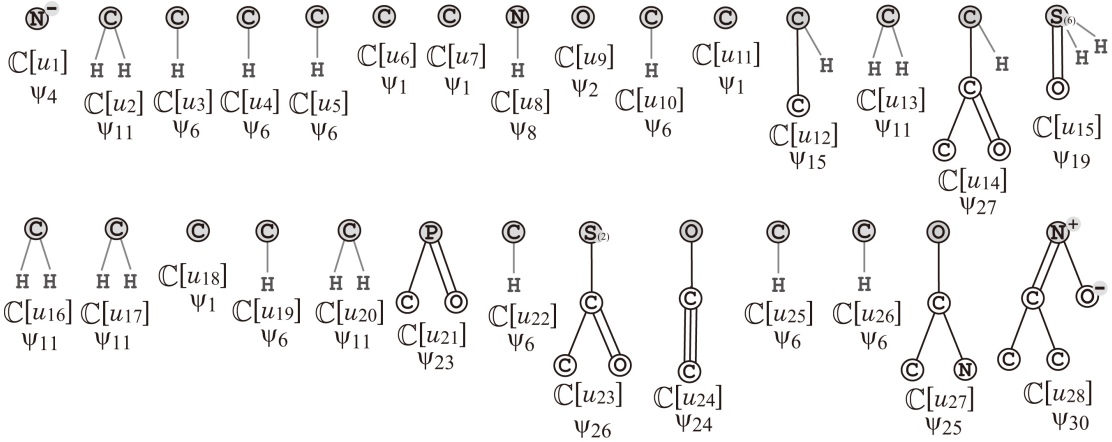


Fig. 3. The set $\mathbb{C}[u_i], i \in [1, 28]$ of the example $\mathbb{C}$ in Fig. 2, where the root of each tree is depicted with a gray circle and the hydrogens attached to non-root vertices are omitted in the figure.

For an integer K, a feature vector $f(\mathbb{C})$ of a chemical graph $\mathbb{C}$ is defined by a *feature function* f that consists of K descriptors. We call $\mathbb{R}^K$ the *feature space*.

Tanaka *et al*. [25] defined a feature vector $f(\mathbb{C}) \in \mathbb{R}^K$ to be a combination of the frequency of edge-configurations of the interior-edges and the frequency of chemical rooted trees among the set of chemical rooted trees $\mathbb{C}[u]$ over all interior-vertices u. In this paper, we introduce the rank and the adjacency-configuration of leaf-edges as new descriptors in a feature vector of a chemical graph. See **Supplementary Material** for a full description of descriptors used in Stage 2 of the framework.

Topological Specification

A topological specification is described as a set of the following rules proposed by Shi *et al*. [26] and modified by Tanaka *et al*. [25]:

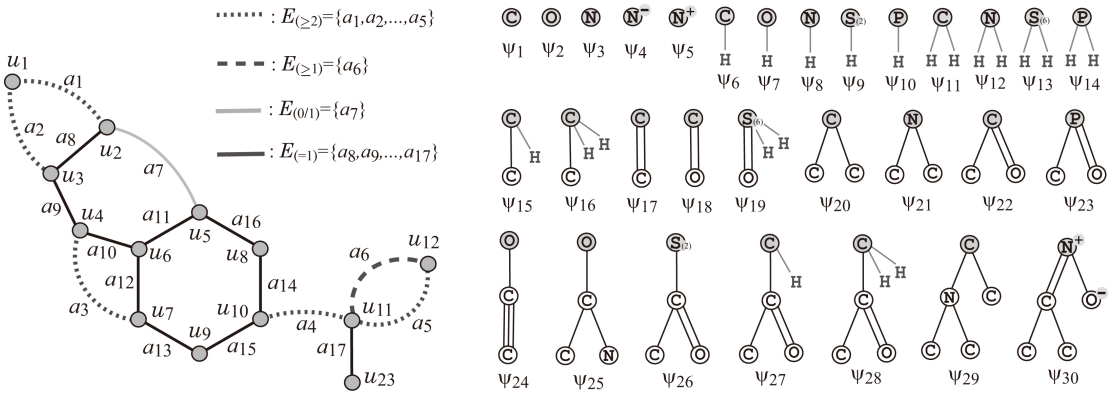
(i) a *seed graph* G_{C} as an abstract form of a target chemical graph $\mathbb{C}$;

(ii) a set $\mathcal{F}$ of chemical rooted trees as candidates for a tree $\mathbb{C}[u]$ rooted at each exterior-vertex u in $\mathbb{C}$; and

(iii) lower and upper bounds on the number of components in a target chemical graph such as chemical elements, double/triple bounds and the interior-vertices in $\mathbb{C}$.

Fig. 4a,b illustrate examples of a seed graph G_{C} and a set $\mathcal{F}$ of chemical rooted trees, respectively. Given a seed graph G_{C}, the interior of a target chemical graph $\mathbb{C}$ is constructed from G_{C} by replacing some edges $a = uv$ with paths P_a between the end-vertices u and v and by attaching new paths Q_v to some vertices v. For example, a chemical graph $\langle\mathbb{C}\rangle$ in Fig. 2 is constructed from the seed graph G_{C} in Fig. 4a as follows.

- First replace five edges $a_1 = u_1u_2, a_2 = u_1u_3, a_3 = u_4u_7, a_4 = u_{10}u_{11}$ and $a_5 = u_{11}u_{12}$ in G_{C} with new paths $P_{a_1} = (u_1, u_{13}, u_2)$, $P_{a_2} = (u_1, u_{14}, u_3)$, $P_{a_3} = (u_4, u_{15}, u_{16}, u_7)$, $P_{a_4} = (u_{10}, u_{17}, u_{18}, u_{19}, u_{11})$ and



(a) A seed graph G_C=(V_C,E_C) (b) A set $\mathcal{F}$ of chemical rooted trees

Fig. 4. (a) An illustration of a seed graph G_{C} with $\mathrm{r}(G_{\mathrm{C}}) = 5$ where the vertices in V_{C} are depicted with gray circles, the edges in $E_{(\geq 2)}$ are depicted with dotted lines, the edges in $E_{(\geq 1)}$ are depicted with dashed lines, the edges in $E_{(0/1)}$ are depicted with gray bold lines and the edges in $E_{(=1)}$ are depicted with black solid lines; (b) A set $\mathcal{F} = \{\psi_1, \psi_2, \ldots, \psi_{30}\} \subseteq \mathcal{F}(D_\pi)$ of 30 chemical rooted trees $\psi_i, i \in [1, 30]$, where the root of each tree is depicted with a gray circle, where the hydrogens attached to non-root vertices are omitted in the figure.

$P_{a_5} = (u_{11}, u_{20}, u_{21}, u_{22}, u_{12})$, respectively to obtain a subgraph G_1 of $\langle \mathbb{C} \rangle$.

- Next attach to this graph G_1 three new paths $Q_{u_5} = (u_5, u_{24})$, $Q_{u_{18}} = (u_{18}, u_{25}, u_{26}, u_{27})$ and $Q_{u_{22}} = (u_{22}, u_{28})$ to obtain the interior of $\langle \mathbb{C} \rangle$ in Fig. 2.
- Finally attach to the interior 28 trees selected from the set $\mathcal{F}$ and assign chemical elements and bond-multiplicities in the interior to obtain a chemical graph $\mathbb{C}$ in Fig. 2. In Fig, 3, $\psi_1 \in \mathcal{F}$ is selected for $\mathbb{C}[u_i]$, $i \in \{6, 7, 11\}$. Similarly ψ_2 for $\mathbb{C}[u_9]$, ψ_4 for $\mathbb{C}[u_1]$, ψ_6 for $\mathbb{C}[u_i]$, $i \in \{3, 4, 5, 10, 19, 22, 25, 26\}$, ψ_8 for $\mathbb{C}[u_8]$, ψ_{11} for $\mathbb{C}[u_i]$, $i \in \{2, 13, 16, 17, 20\}$, ψ_{15} for $\mathbb{C}[u_{12}]$, ψ_{19} for $\mathbb{C}[u_{15}]$, ψ_{23} for $\mathbb{C}[u_{21}]$, ψ_{24} for $\mathbb{C}[u_{24}]$, ψ_{25} for $\mathbb{C}[u_{27}]$, ψ_{26} for $\mathbb{C}[u_{23}]$, ψ_{27} for $\mathbb{C}[u_{14}]$ and ψ_{30} for $\mathbb{C}[u_{28}]$.

Our definition of a topological specification is analogous with the one by Tanaka *et al.* [25] except for a necessary modification due to the introduction of multiple valences of chemical elements, cations and anions (see **Supplementary Material** for a full description of topological specification).

4. Linear Regressions

For an integer $p \geq 1$ and a vector $x \in \mathbb{R}^p$, the j-th entry of x is denoted by $x(j), j \in [1, p]$.

Let D be a data set of chemical graphs $\mathbb{C}$ with an observed value $a(\mathbb{C}) \in \mathbb{R}$, where we denote by $a_i = a(\mathbb{C}_i)$ for an indexed graph $\mathbb{C}_i$.

Let f be a feature function that maps a chemical graph $\mathbb{C}$ to a vector $f(\mathbb{C}) \in \mathbb{R}^K$ where we denote by $x_i = f(\mathbb{C}_i)$ for an indexed graph $\mathbb{C}_i$. For a prediction function $\eta : \mathbb{R}^K \to \mathbb{R}$, define an error function

$$\mathrm{Err}(\eta; D) \triangleq \sum_{\mathbb{C}_i \in D} (a_i - \eta(f(\mathbb{C}_i)))^2 = \sum_{\mathbb{C}_i \in D} (a_i - \eta(x_i))^2, \quad (4)$$

and define the *coefficient of determination* $R^2(\eta, D)$ to be

$$\mathrm{R}^2(\eta, D) \triangleq 1 - \frac{\mathrm{Err}(\eta; D)}{\sum_{\mathbb{C}_i \in D} (a_i - \widetilde{a})^2} \text{ for } \widetilde{a} = \frac{1}{|D|} \sum_{\mathbb{C} \in D} a(\mathbb{C}). \quad (5)$$

For a feature space $\mathbb{R}^K$, a hyperplane is denoted by a pair (w, b) of a vector $w \in \mathbb{R}^K$ and a real $b \in \mathbb{R}$. Given a hyperplane $(w, b) \in \mathbb{R}^K w$, a prediction function $\eta_{w,b} : \mathbb{R}^K \to \mathbb{R}$ is defined by setting

$$\eta_{w,b}(x) \triangleq w \cdot x + b = \sum_{j \in [1,K]} w(j)x(j) + b. \quad (6)$$

We wish to find a hyperplane (w, b) that minimizes the error function $\mathrm{Err}(\eta_{w,b}; D)$. In many cases, a feature vector f contains descriptors that do not play an essential role in constructing a good prediction function. When we solve the minimization problem, the entries $w(j)$ for some descriptors $j \in [1, K]$ in the resulting hyperplane (w, b) become zero, which means that these descriptors were not necessarily important for finding a prediction function $\eta_{w,b}$. It is proposed that solving the minimization with an additional penalty term τ to the error function often results in more number of entries $w(j) = 0$, reducing a set of descriptors necessary for defining a prediction function $\eta_{w,b}$. For an error function with such a penalty term, a Ridge function $\frac{1}{2|D|} \mathrm{Err}(\eta_{w,b}; D) + \lambda \left[\sum_{j \in [1,K]} w(j)^2 + b^2\right]$ [28] and a Lasso function

$\frac{1}{2|D|}\,\mathrm{Err}\,(\eta_{w,b};D) + \lambda\left[\sum_{j\in[1,K]}|w(j)|+|b|\right]$ [29] are known, where $\lambda \in \mathbb{R}$ is a given real number.

Given a prediction function $\eta_{w,b}$, we can simulate a process of computing the output $\eta_{w,b}(x)$ for an input $x \in \mathbb{R}^K$ as an MILP $\mathcal{M}(x,y;\mathcal{C}_1)$ in the framework. By solving such an MILP for a specified target value y^*, we can find a vector $x^* \in \mathbb{R}^K$ such that $\eta_{w,b}(x^*) = y^*$. Instead of specifying a single target value y^*, we use lower and upper bounds $\underline{y}^*, \overline{y}^* \in \mathbb{R}$ on the value $a(\mathbb{C})$ of a chemical graph $\mathbb{C}$ to be inferred. We can control the range between $\underline{y}^*$ and $\overline{y}^*$ for searching a chemical graph $\mathbb{C}$ by setting $\underline{y}^*$ and $\overline{y}^*$ to be close or different values. A desired MILP is formulated as follows.

$\mathcal{M}(x,y;\mathcal{C}_1)$: An MILP formulation for the inverse problem to prediction function .

constants:

- A hyperplane (w,b) with $w \in \mathbb{R}^K$ and $b \in \mathbb{R}$;
- Real values $\underline{y}^*, \overline{y}^* \in \mathbb{R}$ such that $\underline{y}^* < \overline{y}^*$;
- A set $I_{\mathbb{Z}}$ of indices $j \in [1,K]$ such that the j-th descriptor $\mathrm{dcp}_j(\mathbb{C})$ is always an integer;
- A set I_+ of indices $j \in [1,K]$ such that the j-th descriptor $\mathrm{dcp}_j(\mathbb{C})$ is always non-negative;
- $\ell(j), u(j) \in \mathbb{R}$, $j \in [1,K]$: lower and upper bounds on the j-th descriptor;

variables:

- Non-negative integer variable $x(j) \in \mathbb{Z}_+$, $j \in I_{\mathbb{Z}} \cap I_+$;
- Integer variable $x(j) \in \mathbb{Z}$, $j \in I_{\mathbb{Z}} \setminus I_+$;
- Non-negative real variable $x(j) \in \mathbb{R}_+$, $j \in I_+ \setminus I_{\mathbb{Z}}$;
- Real variable $x(j) \in \mathbb{R}$, $j \in [1,K] \setminus (I_{\mathbb{Z}} \cup I_+)$;

constraints:

$$\begin{aligned} &\ell(j) \le x(j) \le u(j), j \in [1,K]; \\ &\underline{y}^* \le \sum_{j\in[1,K]} w(j)x(j) + b \le \overline{y}^* \end{aligned} \tag{7}$$

objective function:
none.

The number of variables and constraints in the above MILP formulation is $O(K)$. It is not difficult to see that the above MILP is an NP-hard problem. The entire MILP for Stage 4 consists of the two MILPs $\mathcal{M}(x,y;\mathcal{C}_1)$ and $\mathcal{M}(g,x;\mathcal{C}_2)$ with no objective function. The latter represents the computation process of our feature function f and a given topological specification. See **Supplementary Material** for the details of MILP $\mathcal{M}(g,x;\mathcal{C}_2)$.

5. Results

We implemented our method of Stages 1 to 5 for inferring chemical graphs under a given topological specification and conducted experiments to evaluate the computational efficiency. We executed the experiments on a PC with Processor: Core i7-9700 (3.0 GHz; 4.7 GHz at the maximum) and Memory: 16 GB RAM DDR4.

Results on Phase 1. We have conducted experiments of linear regression for 37 chemical properties among which we report the following 18 properties to which the test coefficient of determination R^2 attains at least 0.8: octanol/water partition coefficient (Kow), heat of combustion (Hc), vapor density (Vd), optical rotation (OptR), electron density on the most positive atom (EDPA), melting point (Mp), heat of atomization (Ha), heat of formation (Hf), internal energy at 0K (U0), energy of lowest unoccupied molecular orbital (Lumo), isotropic polarizability (Alpha), heat capacity at 298.15K (Cv), solubility (Sl), surface tension (SfT), viscosity (Vis), isobaric heat capacities in liquid phase (IhcLiq), isobaric heat capacities in solid phase (IhcSol) and lipophilicity (Lp).

We used data sets provided by HSDB from PubChem [30] for Kow, Hc, Vd and OptR, M. Jalali-Heravi and M. Fatemi [31] for EDPA, Roy and Saha [32] for Mp, Ha and Hf, MoleculeNet [33] for U0, Lumo, Alpha, Cv and Sl, Goussard *et al.* [34] for SfT and Vis, R. Naef [35] for IhcLiq and IhcSol, and Figshare [36] for Lp.

Properties U0, Lumo, Alpha and Cv share a common original data set D^* with more than 130,000 compounds, and we used a set D_π of 1,000 graphs randomly selected from D^* as a common data set of these four properties π in this experiment.

Stages 1, 2 and 3 in Phase 1 are implemented as follows.

Stage 1. We set a graph class $\mathcal{G}$ to be the set of all chemical graphs with any graph structure, and set a branch-parameter ρ to be 2.

For each of the properties, we first select a set Λ of chemical elements and then collect a data set D_π on chemical graphs over the set Λ of chemical elements. During construction of the data set D_π, chemical compounds that do not satisfy one of the following are eliminated: the graph is connected, the number of non-hydrogen neighbors of each atom is at most four, and the number of carbon atoms is at least four.

Table 1 shows the size and range of data sets that we prepared for each chemical property in Stage 1, where we denote the following:

- Λ: the set of elements used in the data set D_π; Λ is one of the following 11 sets: $\Lambda_1 = \{\mathrm{H},\mathrm{C},\mathrm{O}\}$; $\Lambda_2 = \{\mathrm{H},\mathrm{C},\mathrm{O},\mathrm{N}\}$; $\Lambda_3 = \{\mathrm{H},\mathrm{C},\mathrm{O},\mathrm{S}_{(2)}\}$; $\Lambda_4 = \{\mathrm{H},\mathrm{C},\mathrm{O},\mathrm{Si}\}$; $\Lambda_5 = \{\mathrm{H},\mathrm{C},\mathrm{O},\mathrm{N},\mathrm{Cl},\mathrm{P}_{(3)},\mathrm{P}_{(5)}\}$; $\Lambda_6 = \{\mathrm{H},\mathrm{C},\mathrm{O},\mathrm{N},\mathrm{S}_{(2)},\mathrm{F}\}$; $\Lambda_7 = \{\mathrm{H},\mathrm{C},\mathrm{O},\mathrm{N},\mathrm{S}_{(2)},\mathrm{S}_{(6)},\mathrm{Cl}\}$; $\Lambda_8 = \{\mathrm{H},\mathrm{C}_{(2)},\mathrm{C}_{(3)},\mathrm{C}_{(4)},\mathrm{O},\mathrm{N}_{(2)},\mathrm{N}_{(3)}\}$; $\Lambda_9 = \{\mathrm{H},\mathrm{C},\mathrm{O},\mathrm{N},\mathrm{S}_{(2)},\mathrm{S}_{(4)},\mathrm{S}_{(6)},\mathrm{Cl}\}$; $\Lambda_{10} = \{\mathrm{H},\mathrm{C}_{(2)},\mathrm{C}_{(3)},\mathrm{C}_{(4)},\mathrm{C}_{(5)},\mathrm{O},\mathrm{N}_{(1)},\mathrm{N}_{(2)},\mathrm{N}_{(3)},\mathrm{F}\}$; and $\Lambda_{11} = \{\mathrm{H},\mathrm{C}_{(2)},\mathrm{C}_{(3)},\mathrm{C}_{(4)},\mathrm{O},\mathrm{N}_{(2)},\mathrm{N}_{(3)},\mathrm{S}_{(2)},\mathrm{S}_{(4)},\mathrm{S}_{(6)},\mathrm{Cl}\}$, where $\mathsf{e}_{(i)}$ for a chemical element e and an integer $i \ge 1$ means that a chemical element e with valence i.

Table 1. Results in Phase 1.

π	Λ	$\lvert D_\pi\rvert$	$\underline{n}$, $\overline{n}$	$\underline{a}$, $\overline{a}$	$\lvert\Gamma\rvert$	$\lvert\mathcal{F}\rvert$	K	λ_π	K'	test R^2
Kow	Λ_2	684	4, 58	–7.5, 15.6	25	166	223	6.4E−5	80.3	0.953
Kow	Λ_9	899	4, 69	–7.5, 15.6	37	219	303	5.5E−5	112.1	0.927
Hc	Λ_2	255	4, 63	49.6, 35099.6	17	106	154	1.9E−4	19.2	0.946
Hc	Λ_7	282	4, 63	49.6, 35099.6	21	118	177	1.9E−4	20.5	0.951
Vd	Λ_2	474	4, 30	0.7, 20.6	21	160	214	1.0E−3	3.6	0.927
Vd	Λ_5	551	4, 30	0.7, 20.6	24	191	256	5.5E−4	8.0	0.942
OptR	Λ_2	147	5, 44	–117.0, 165.0	21	55	107	4.6E−4	39.2	0.823
OptR	Λ_6	157	5, 69	–117.0, 165.0	25	62	123	7.3E−4	41.7	0.825
EDPA	Λ_1	52	11, 16	0.80, 3.76	9	33	64	1.0E−4	10.9	0.999
Mp	Λ_2	467	4, 122	–185.33, 300.0	23	142	197	3.7E−5	82.5	0.817
Ha	Λ_3	115	4, 11	1100.6, 3009.6	8	83	115	3.7E−5	39.0	0.997
Hf	Λ_1	82	4, 16	30.2, 94.8	5	50	74	1.0E−4	34.0	0.987
U0	Λ_{10}	977	4, 9	–570.6, –272.8	59	190	297	1.0E−7	246.7	0.999
Lumo	Λ_{10}	977	4, 9	–0.11, 0.10	59	190	297	6.4E−5	133.9	0.841
Alpha	Λ_{10}	977	4, 9	50.9, 99.6	59	190	297	1.0E−5	125.5	0.961
Cv	Λ_{10}	977	4, 9	19.2, 44.0	59	190	297	1.0E−5	165.3	0.961
Sl	Λ_9	915	4, 55	–11.6, 1.11	42	207	300	7.3E−5	130.6	0.808
SfT	Λ_4	247	5, 33	12.3, 45.1	11	91	128	6.4E−4	20.9	0.804
Vis	Λ_4	282	5, 36	–0.64, 1.63	12	88	126	8.2E−4	16.3	0.893
IhcLiq	Λ_2	770	4, 78	106.3, 1956.1	23	200	256	1.9E−5	82.2	0.987
IhcLiq	Λ_7	865	4, 78	106.3, 1956.1	29	246	316	8.2E−6	139.1	0.986
IhcSol	Λ_8	581	5, 70	67.4, 1220.9	33	124	192	2.8E−5	75.9	0.985
IhcSol	Λ_{11}	668	5, 70	67.4, 1220.9	40	140	228	2.8E−5	86.7	0.982
Lp	Λ_2	615	6, 60	–3.62, 6.84	32	116	186	1.0E−4	98.5	0.856
Lp	Λ_9	936	6, 74	–3.62, 6.84	44	136	231	6.4E−5	130.4	0.840

- $|D_\pi|$: the size of data set D_π over Λ for the property π.
- $\underline{n}$, $\overline{n}$: the minimum and maximum values of the number $n(\mathbb{C})$ of non-hydrogen atoms in compounds $\mathbb{C}$ in D_π.
- $\underline{a}$, $\overline{a}$: the minimum and maximum values of $a(\mathbb{C})$ for π over compounds $\mathbb{C}$ in D_π.
- $|\Gamma|$: the number of different edge-configurations of interior-edges over the compounds in D_π.
- $|\mathcal{F}|$: the number of non-isomorphic chemical rooted trees in the set of all 2-fringe-trees in the compounds in D_π.
- K: the number of descriptors in a feature vector $f(\mathbb{C})$.

Stage 2. The newly defined feature function in our chemical model without suppressing hydrogen in Section 3 is used. We standardize the range of each descriptor and the range $\{t \in \mathbb{R} \mid \underline{a} \leq t \leq \bar{a}\}$ of property values $a(\mathbb{C}), \mathbb{C} \in D_\pi$.

Stage 3. For each chemical property π, we select a penalty value λ_π in the Lasso function from 36 different values from 0 to 100 by conducting linear regression as a preliminary experiment.

We conducted an experiment in Stage 3 to evaluate the performance of the prediction function based on cross-validation. For a property π, an execution of a *cross-validation* consists of five trials of constructing a prediction function as follows. First partition the data set D_π into five subsets $D^{(k)}$, $k \in [1,5]$ randomly. For each $k \in [1,5]$, the k-th trial constructs a prediction function $\eta^{(k)}$ by conducting a linear regression with the penalty term λ_π using the set $D_\pi \setminus D^{(k)}$ as a training data set. We used scikit-learn version 0.23.2 with Python 3.8.5 for executing linear regression with Lasso function. For each property, we executed ten cross-validations and we show the median of test $R^2(\eta^{(k)}, D^{(k)}), k \in [1,5]$ over all ten cross-validations. Recall that a subset of descriptors is selected in linear regression with Lasso function and let K' denote the average number of selected descriptors over all 50 trials. The running time per trial in a cross-validation was at most one second.

Table 1 shows the results on Stages 2 and 3, where we denote the following:

- λ_π: the penalty value in the Lasso function selected for a property π, where aEb means $a \times 10^b$.
- K': the average of the number of descriptors selected in the linear regression over all 50 trials in ten cross-validations.
- test R^2: the median of test R^2 over all 50 trials in ten cross-validations.

Recall that the adjacency-configuration for leaf-edges was introduced as a new descriptor in this paper. Without including this new descriptor, the test R^2 for property Vis was 0.790, that for Lumo was 0.799 and that for Mp was

0.796, while the test R^2 for each of the other properties in Table 1 was almost the same.

From Table 1, we observe that a relatively large number of properties admit a good prediction function based on linear regression. The number K' of descriptors used in linear regression is considerably small for some properties. For example of property VD, the four descriptors most frequently selected in the case of $\Lambda = \{\mathrm{H}, \mathrm{O}, \mathrm{C}, \mathrm{N}\}$ are the number of non-hydrogen atoms; the number of interior-vertices v with $\deg_{\mathrm{C^{int}}}(v) = 1$; the number of fringe-trees r-isomorphic to the chemical rooted tree ψ_1 in Fig. 5; and the number of leaf-edges with adjacency-configuration $(\mathrm{O}, \mathrm{C}, 2)$. The eight descriptors most frequently selected in the case of $\Lambda = \{\mathrm{H}, \mathrm{O}, \mathrm{C}, \mathrm{N}, \mathrm{Cl}, \mathrm{P}_{(3)}, \mathrm{P}_{(5)}\}$ are the number of non-hydrogen atoms; the number of interior-vertices v with $\deg_{\mathrm{C^{int}}}(v) = 1$; the number of exterior-vertices v with $\alpha(v) = \mathrm{Cl}$; the number of interior-edges with edge-configuration $\gamma_i, i = 1, 2$, where $\gamma_1 = (\mathrm{C2}, \mathrm{C2}, 2)$ and $\gamma_2 = (\mathrm{C3}, \mathrm{C4}, 1)$; and the number of fringe-trees r-isomorphic to the chemical rooted tree $\psi_i, i = 1, 2, 3$ in Fig. 5.

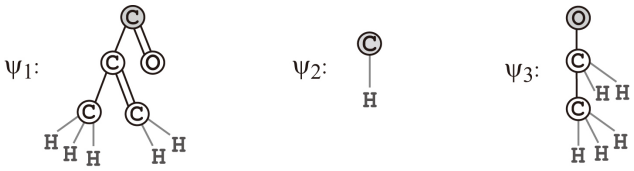


Fig. 5. An illustration of chemical rooted trees ψ_1, ψ_1 and ψ_3 that are selected in Lasso linear regression for constructing a prediction function to property VD, where the root is depicted with a gray circle.

For the 18 properties listed in Table 1, we used ANN to construct prediction functions. For this purpose, we used our newly proposed feature vector and the experimental setup as explained in Tanaka *et al.* [25]. From these computation experiments, we observe that for the properties HC, VD, HA, HF, U0, ALPHA and CV, the test R^2 scores of the prediction functions obtained by Lasso linear regression is at least 0.05 more than those obtained by ANN. For the properties OPTR, SL and SFT, the test R^2 scores of the prediction functions obtained by ANN is at least 0.05 more than those obtained by Lasso linear regression. For the other properties, the test R^2 scores obtained by Lasso linear regression and ANN are comparable.

Results on Phase 2. We used a set of seven instances I_{a}, $I_{\mathrm{b}}^i, i \in [1, 4]$, I_{c} and I_{d} based on seed graphs prepared by Shi *et al.* [26] to execute Stages 4 and 5 in Phase 2. We here present their seed graphs G_{C} (see **Supplementary Material** for the details of instances I_{a}, $I_{\mathrm{b}}^i, i \in [1, 4]$, I_{c} and I_{d}). The seed graph G_{C} of instance I_{a} is illustrated in Fig. 4a. The seed graph G_{C}^1 (resp., $G_{\mathrm{C}}^i, i = 2, 3, 4$) of instances I_{b}^1 and I_{d} (resp., $I_{\mathrm{b}}^i, i = 2, 3, 4$) is illustrated in Fig. 6.

Instance I_{c} has been introduced by Shi *et al.* [26] in order to infer a chemical graph $\mathbb{C}^\dagger$ such that the core of $\mathbb{C}^\dagger$ is the same as the core of chemical graph $\mathbb{C}_A$: CID 24822711 in Fig. 7a and the frequency of each edge-configuration in the non-core of $\mathbb{C}^\dagger$ is the same as that of chemical graph $\mathbb{C}_B$: CID 59170444 illustrated in Fig. 7b. This means that the seed graph G_{C} of I_{c} is the core of $\mathbb{C}_A$ which is indicated by a shaded area in Fig. 7a.

Instance I_{d} has been introduced by Shi *et al.* [26] in order to infer a monocyclic chemical graph $\mathbb{C}^\dagger$ such that the frequency vector of edge-configurations in $\mathbb{C}^\dagger$ is a vector obtained by merging those of two chemical graphs $\mathbb{C}_A$: CID 10076784 and $\mathbb{C}_B$: CID 44340250 illustrated in Fig. 7c,d, respectively.

Stage 4. We executed Stage 4 for five properties $\pi \in \{$HC, VD, OPTR, IHCLIQ, VIS$\}$.

For the MILP formulation $\mathcal{M}(x, y; \mathcal{C}_1)$ in Section 4, we use the prediction function $\eta_{w,b}$ that attained the median test R^2 in Table 1. We used CPLEX version 12.10 to solve an MILP in Stage 4. Tables 2,3,4,5,6 show the computational results of the experiment in Stage 4 for the five properties, where we denote the following:

- $\underline{y}^*$, $\overline{y}^*$: lower and upper bounds $\underline{y}^*$, $\overline{y}^* \in \mathbb{R}$ 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 in Stage 4;
- I-time: the time (sec.) to solve the MILP in Stage 4;
- n: the number $n\left(\mathbb{C}^\dagger\right)$ of non-hydrogen atoms in the chemical graph $\mathbb{C}^\dagger$ inferred in Stage 4; and
- $\mathrm{n^{int}}$: the number $\mathrm{n^{int}}\left(\mathbb{C}^\dagger\right)$ of interior-vertices in the chemical graph $\mathbb{C}^\dagger$ inferred in Stage 4;
- $\eta(f(\mathbb{C}^\dagger))$: the predicted property value $\eta(f(\mathbb{C}^\dagger))$ of the chemical graph $\mathbb{C}^\dagger$ inferred in Stage 4.

From Tables 2,3,4,5,6 we observe that an instance with a large number of variables and constraints takes more running time than those with a smaller size in general. We solved all instances in this experiment with our MILP formulation in a few seconds to around 30 seconds.

Fig. 8a–e illustrate the chemical graphs $\mathbb{C}^\dagger$ inferred from I_{c} with $(\underline{y}^*, \overline{y}^*) = (13700, 13800)$ of HC, I_{b}^2 with $(\underline{y}^*, \overline{y}^*) = (21, 22)$ of VD, I_{b}^4 with $(\underline{y}^*, \overline{y}^*) = (70, 71)$ of OPTR, I_{d} with $(\underline{y}^*, \overline{y}^*) = (1190, 1210)$ of IHCLIQ, and I_{b}^3 with $(\underline{y}^*, \overline{y}^*) = (1.85, 1.90)$ of VIS, respectively.

Similarly, we executed Stage 4 for these seven instances I_{a}, $I_{\mathrm{b}}^i, i \in [1, 4]$, I_{c} and I_{d} for five properties $\pi \in \{$HC, VD, OPTR, IHCLIQ, VIS$\}$ by using the prediction functions obtained by ANN. We list the running time to solve MILP formulation for each of these instances in Tables 7,8. From the computation experiments, we observe that for many instances, the running time is significantly faster than that of Stage 4 based on ANN.

Inferring a chemical graph with target values in multiple properties Once we obtained prediction functions η_π for several properties π, include MILP formulations for these functions η_π into a single MILP $\mathcal{M}(x, y; \mathcal{C}_1)$ so as

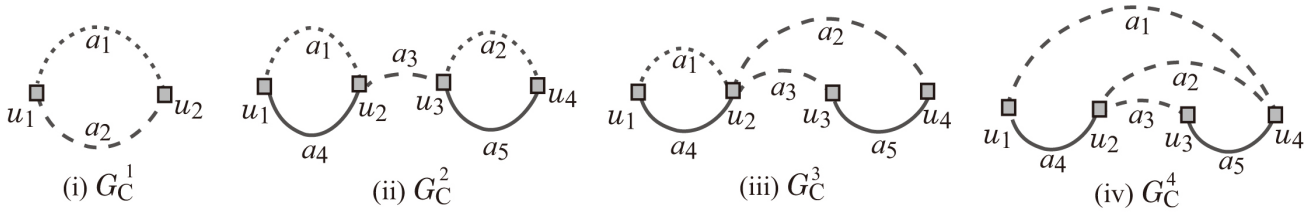
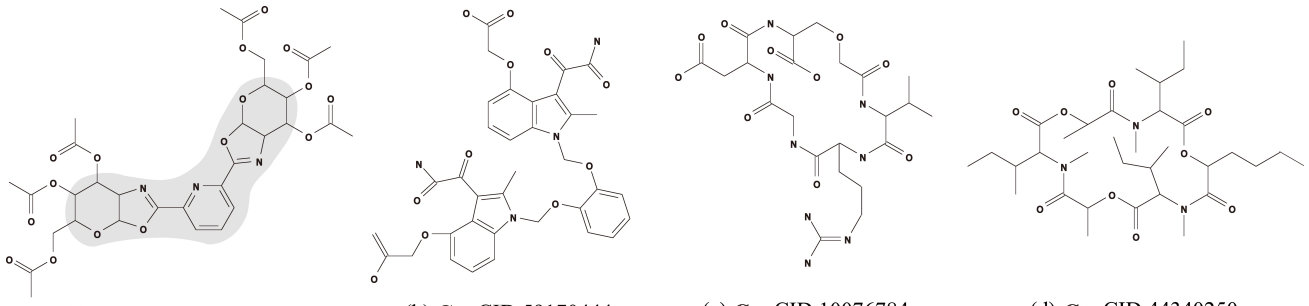


Fig. 6. (i) Seed graph G_C^1 for I_b^1 and I_d; (ii) Seed graph G_C^2 for I_b^2; (iii) Seed graph G_C^3 for I_b^3; (iv) Seed graph G_C^4 for I_b^4.



(a) $\mathbb{C}_A$: CID 24822711 (b) $\mathbb{C}_B$: CID 59170444 (c) $\mathbb{C}_A$: CID 10076784 (d) $\mathbb{C}_B$: CID 44340250

Fig. 7. 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.

Table 2. Results of Stages 4 and 5 for HC using Lasso linear regression.

inst.	$\underline{y}^*$, $\overline{y}^*$	#v	#c	I-time	n	n$^{\text{int}}$	$\eta(f(\mathbb{C}^\dagger))$	D-time	$\mathbb{C}$-LB	#$\mathbb{C}$
I_a	5950, 6050	9902	9255	4.6	44	25	5977.9	0.068	1	1
I_b^1	5950, 6050	9404	6776	1.7	36	10	6007.1	0.048	6	6
I_b^2	5950, 6050	11729	9891	16.7	50	25	6043.7	38.7	2.4×10^5	100
I_b^3	5950, 6050	11510	9894	16.3	47	25	6015.4	0.353	8724	100
I_b^4	5950, 6050	11291	9897	9.0	49	26	5971.6	0.304	84	84
I_c	13700, 13800	6915	7278	0.7	50	33	13703.3	0.016	1	1
I_d	13700, 13800	5535	6781	4.9	44	23	13704.7	0.564	4.3×10^5	100

Table 3. Results of Stages 4 and 5 for VD using Lasso linear regression.

inst.	$\underline{y}^*$, $\overline{y}^*$	#v	#c	I-time	n	n$^{\text{int}}$	$\eta(f(\mathbb{C}^\dagger))$	D-time	$\mathbb{C}$-LB	#$\mathbb{C}$
I_a	16, 17	9481	9358	1.6	38	23	16.83	0.070	1	1
I_b^1	16, 17	9928	6986	1.5	35	12	16.68	0.206	48	48
I_b^2	21, 22	12373	10101	10.0	48	25	21.62	0.104	20	20
I_b^3	21, 22	12159	10104	6.5	48	25	21.95	3.65	8.6×10^5	100
I_b^4	21, 22	11945	10107	8.1	48	25	21.34	0.057	6	6
I_c	21, 22	7073	7438	0.7	50	34	21.89	0.016	1	1
I_d	17, 18	5693	6942	2.1	41	23	17.94	0.161	216	100

Table 4. Results of Stages 4 and 5 for OPTR using Lasso linear regression.

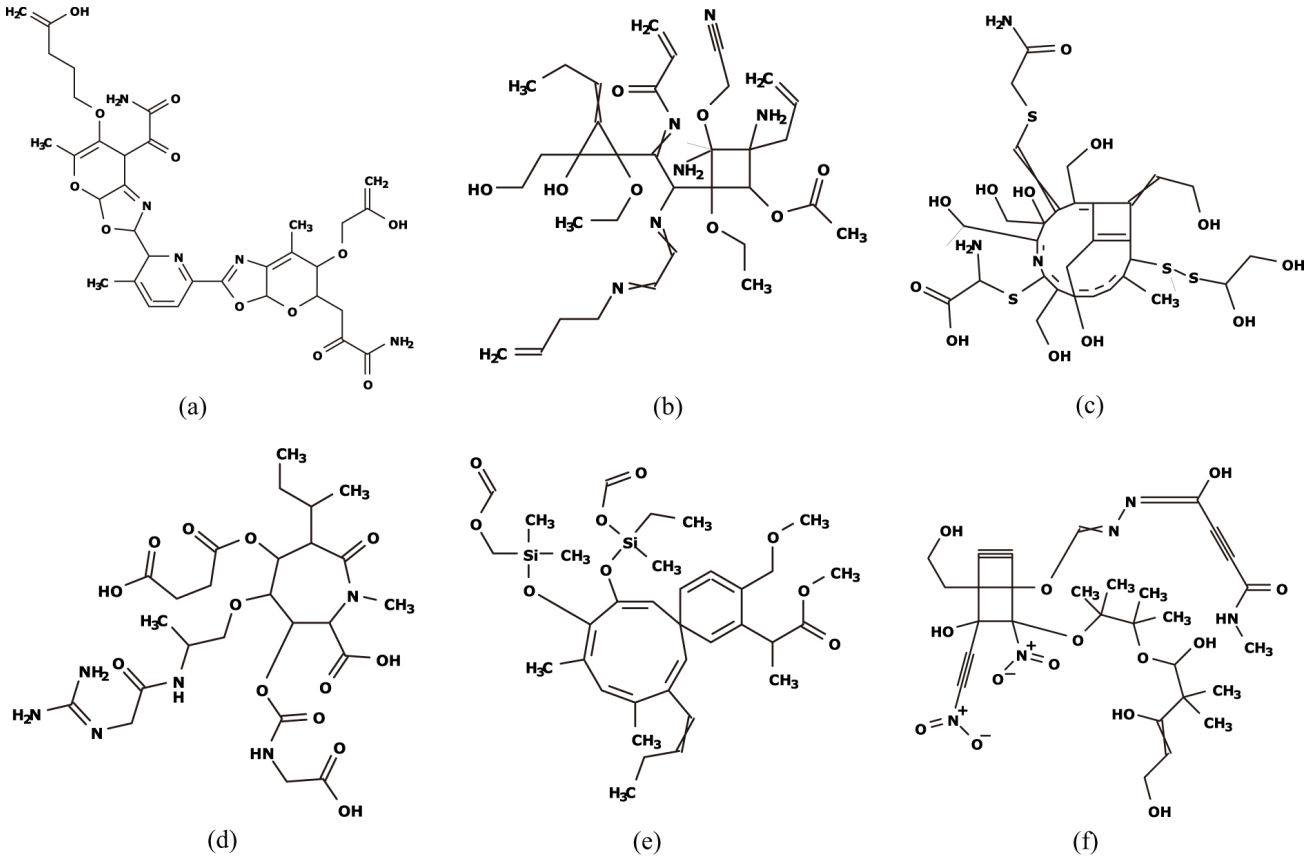
inst.	$\underline{y}^*$, $\overline{y}^*$	#v	#c	I-time	n	n$^{\text{int}}$	$\eta(f(\mathbb{C}^\dagger))$	D-time	$\mathbb{C}$-LB	#$\mathbb{C}$
I_a	70, 71	8962	9064	3.5	40	23	70.1	0.061	1	1
I_b^1	70, 71	9432	6662	2.7	37	14	70.1	0.185	2622	100
I_b^2	70, 71	11818	9773	10.0	50	25	70.8	0.041	4	4
I_b^3	70, 71	11602	9776	10.2	50	25	70.2	0.241	60	60
I_b^4	70, 71	11386	9779	24.7	49	25	70.9	6.39	4.6×10^5	100
I_c	–112, –111	6807	7170	1.8	50	32	-111.9	0.016	1	1
I_d	70, 71	5427	6673	6.1	42	23	70.2	0.127	78768	100

Table 5. Results of Stages 4 and 5 for IhcLiq using Lasso linear regression.

inst.	$\underline{y}^*$, $\overline{y}^*$	#v	#c	I-time	n	n$^{\text{int}}$	$\eta(f(\mathbb{C}^\dagger))$	D-time	$\mathbb{C}$-LB	#$\mathbb{C}$
I_{a}	1190, 1210	10180	9538	3.9	48	26	1208.5	0.071	2	2
I_{b}^1	1190, 1210	10784	7191	2.4	35	14	1206.7	0.082	12	12
I_{b}^2	1190, 1210	13482	10302	14.1	47	25	1206.7	0.11	12	12
I_{b}^3	1190, 1210	13275	10301	9.0	49	27	1209.9	0.090	24	24
I_{b}^4	1190, 1210	13128	10306	16.5	50	25	1208.4	0.424	2388	100
I_{c}	1190, 1210	7193	7560	0.8	50	33	1196.5	0.016	1	1
I_{d}	1190, 1210	5813	7063	2.2	44	23	1198.8	5.63	5.2×10^5	100

Table 6. Results of Stages 4 and 5 for Vis using Lasso linear regression.

inst.	$\underline{y}^*$, $\overline{y}^*$	#v	#c	I-time	n	n$^{\text{int}}$	$\eta(f(\mathbb{C}^\dagger))$	D-time	$\mathbb{C}$-LB	#$\mathbb{C}$
I_{a}	1.25, 1.30	6847	8906	1.3	38	22	1.295	0.042	2	2
I_{b}^1	1.25, 1.30	7270	6397	2.5	36	15	1.272	0.155	140	100
I_{b}^2	1.85, 1.90	8984	9512	8.9	45	25	1.879	0.149	288	100
I_{b}^3	1.85, 1.90	8741	9515	16.2	45	26	1.880	0.137	4928	100
I_{b}^4	1.85, 1.90	8498	9518	8.1	45	25	1.851	0.13	660	100
I_{c}	2.75, 2.80	6813	7162	1.0	50	33	2.763	0.025	4	4
I_{d}	1.85, 1.90	5433	6665	2.7	41	23	1.881	0.138	4608	100



(a) (b) (c) (d) (e) (f)

Fig. 8. (a) $\mathbb{C}^\dagger$ with $\eta(f(\mathbb{C}^\dagger)) = 13703.3$ inferred from I_{c} with $(\underline{y}^*, \overline{y}^*) = (13700, 13800)$ of Hc; (b) $\mathbb{C}^\dagger$ with $\eta(f(\mathbb{C}^\dagger)) = 21.62$ inferred from I_{b}^2 with $(\underline{y}^*, \overline{y}^*) = (21, 22)$ of Vd; (c) $\mathbb{C}^\dagger$ with $\eta(f(\mathbb{C}^\dagger)) = 70.9$ inferred from I_{b}^4 with $(\underline{y}^*, \overline{y}^*) = (70, 71)$ of OptR; (d) $\mathbb{C}^\dagger$ with $\eta(f(\mathbb{C}^\dagger)) = 1198.8$ inferred from I_{d} with $(\underline{y}^*, \overline{y}^*) = (1190, 1210)$ of IhcLiq; (e) $\mathbb{C}^\dagger$ with $\eta(f(\mathbb{C}^\dagger)) = 1.880$ inferred from I_{b}^3 with $(\underline{y}^*, \overline{y}^*) = (1.85, 1.90)$ of Vis; (f) $\mathbb{C}^\dagger$ inferred from I_{b}^4 with lower and upper bounds on the predicted property value $\eta_\pi(f(\mathbb{C}^\dagger))$ of property $\pi \in \{\text{Kow, Lp, Sl}\}$ in Table 9.

Table 7. Running time of Stage 4 for HC, VD and OPTR using ANN.

HC			VD			OPTR		
inst.	$\underline{y}^*$, $\overline{y}^*$	I-time	inst.	$\underline{y}^*$, $\overline{y}^*$	I-time	inst.	$\underline{y}^*$, $\overline{y}^*$	I-time
I_a	13350, 13450	24.7	I_a	18, 19	18.1	I_a	62, 63	35.6
I_b^1	9650, 9750	13.5	I_b^1	13, 14	9.4	I_b^1	109, 110	15.5
I_b^2	16750, 16850	70.4	I_b^2	15, 16	40.9	I_b^2	23, 24	192.6
I_b^3	12350, 12450	87.0	I_b^3	20, 21	46.3	I_b^3	-2, -1	936.4
I_b^4	14250, 14350	70.9	I_b^4	22, 23	27.1	I_b^4	19, 20	63.9
I_c	10400, 10500	31.3	I_c	20, 21	20.5	I_c	86, 87	16.4
I_d	12500, 12600	44.3	I_d	18, 19	6.1	I_d	30, 31	31.8

Table 8. Running time of Stage 4 for IHCLIQ and VIS using ANN.

IHCLIQ			VIS		
inst.	$\underline{y}^*$, $\overline{y}^*$	I-time	inst.	$\underline{y}^*$, $\overline{y}^*$	I-time
I_a	980, 1000	56.6	I_a	1.85, 1.90	2.0
I_b^1	1000, 1020	40.4	I_b^1	1.95, 2.00	3.5
I_b^2	1130, 1150	71.6	I_b^2	1.85, 1.90	19.7
I_b^3	1240, 1260	45.0	I_b^3	2.35, 2.40	26.0
I_b^4	1240, 1260	105.7	I_b^4	2.50, 2.55	9.3
I_c	810, 830	9.7	I_c	3.90, 3.95	1.8
I_d	1100, 1120	25.8	I_d	3.30, 3.35	8.3

to infer a chemical graph that satisfies given target values y^* for these properties at the same time. As an additional experiment in Stage 4, we inferred a chemical graph that has a desired predicted value each of three properties KOW, LP and SL, where we used the prediction function η_π for each property $\pi \in \{\text{KOW, LP, SL}\}$ constructed in Stage 3. Table 9 shows the result of Stage 4 for inferring a chemical graph $\mathbb{C}^\dagger$ from instances I_b^2, I_b^3 and I_b^4 with $\Lambda = \{\text{H}, \text{C}, \text{N}, \text{O}, \text{S}_{(2)}, \text{S}_{(6)}, \text{Cl}\}$, where we denote the following:

- π: one of the three properties KOW, LP and SL used in the experiment;
- $\underline{y}^*_\pi$, $\overline{y}^*_\pi$: lower and upper bounds $\underline{y}^*_\pi, \overline{y}^*_\pi \in \mathbb{R}$ on the predicted property value $\eta_\pi(f(\mathbb{C}^\dagger))$ of property $\pi \in \{\text{KOW, LP, SL}\}$ for a chemical graph $\mathbb{C}^\dagger$ to be inferred;
- #v (resp., #c): the number of variables (resp., constraints) in the MILP in Stage 4;
- I-time: the time (sec.) to solve the MILP in Stage 4;
- n: the number $n(\mathbb{C}^\dagger)$ of non-hydrogen atoms in the chemical graph $\mathbb{C}^\dagger$ inferred in Stage 4;
- n^{int}: the number $n^{\text{int}}(\mathbb{C}^\dagger)$ of interior-vertices in the chemical graph $\mathbb{C}^\dagger$ inferred in Stage 4; and
- $\eta_\pi(f(\mathbb{C}^\dagger))$: the predicted property value $\eta_\pi(f(\mathbb{C}^\dagger))$ of property $\pi \in \{\text{KOW, LP, SL}\}$ for the chemical graph $\mathbb{C}^\dagger$ inferred in Stage 4.

Fig. 8f illustrates the chemical graph $\mathbb{C}^\dagger$ inferred from I_b^4 with $(\underline{y}^*_{\pi_1}, \overline{y}^*_{\pi_1}) = (-7.50, -7.40)$, $(\underline{y}^*_{\pi_2}, \overline{y}^*_{\pi_2}) = (-0.70, -0.60)$ and $(\underline{y}^*_{\pi_3}, \overline{y}^*_{\pi_3}) = (-11.4, -11.2)$ for π_1 =KOW, π_2 =LP and π_3 =SL, respectively.

Stage 5. We executed Stage 5 to generate more target chemical graphs $\mathbb{C}^*$, where a chemical graph $\mathbb{C}^*$ is called a *chemical isomer* of a target chemical graph $\mathbb{C}^\dagger$ of a topological specification σ if $f(\mathbb{C}^*) = f(\mathbb{C}^\dagger)$ and $\mathbb{C}^*$ also satisfies the same topological specification σ. We computed chemical isomers $\mathbb{C}^*$ of each target chemical graph $\mathbb{C}^\dagger$ inferred in Stage 4. We executed an algorithm to generate chemical isomers of $\mathbb{C}^\dagger$ up to 100 when the number of all chemical isomers exceeds 100. We can obtain such an algorithm from the dynamic programming proposed by Tanaka *et al.* [25] with a slight modification. The algorithm first decomposes $\mathbb{C}^\dagger$ into a set of acyclic chemical graphs, next replaces each acyclic chemical graph T with another acyclic chemical graph T' that admits the same feature vector as that of T, and finally assembles the resulting acyclic chemical graphs into a chemical isomer $\mathbb{C}^*$ of $\mathbb{C}^\dagger$. Also, a lower bound on the total number of all chemical isomers of $\mathbb{C}^\dagger$ can be computed by the algorithm without generating all of them.

Tables 2,3,4,5,6 show the computational results of the experiment in Stage 5 for the five properties, where we denote the following:

- D-time: the running time (sec.) to execute the dynamic programming algorithm in Stage 5 to compute a lower bound on the number of all chemical isomers $\mathbb{C}^*$ of $\mathbb{C}^\dagger$ and generate all (or up to 100) chemical isomers $\mathbb{C}^*$;
- $\mathbb{C}$-LB: a lower bound on the number of all chemical isomers $\mathbb{C}^*$ of $\mathbb{C}^\dagger$; and
- #$\mathbb{C}$: the number of all (or up to 100) chemical isomers $\mathbb{C}^*$ of $\mathbb{C}^\dagger$ generated in Stage 5.

From Tables 2,3,4,5,6, we observe that for many cases the running time for generating up to 100 target chemical graphs in Stage 5 is less than 0.4 seconds. For some chemical graph $\mathbb{C}^\dagger$, no chemical isomer was found by our algorithm. This is because each acyclic chemical graph in the decomposition of $\mathbb{C}^\dagger$ has no alternative acyclic chemical graph than the original one. On the other hand, some chemical graph $\mathbb{C}^\dagger$ such as the one in I_d in Table 2 admits an extremely large number of chemical isomers $\mathbb{C}^*$. Remember that we know such a lower bound $\mathbb{C}$-LB on the number of chemical isomers without generating all of them.

Table 9. Results of Stage 4 for instances $I_b^i, i = 2, 3, 4$ with specified target values of three properties KOW, LP and SL using Lasso linear regression.

inst.	π	$\underline{y}^*_\pi$, $\overline{y}^*_\pi$	#v	#c	I-time	n	n^{int}	$\eta_\pi(f(\mathbb{C}^\dagger))$
I_b^2	KOW	–7.50, –7.40	14574	11604	62.7	50	30	–7.41
	LP	–1.40, –1.30						–1.33
	SL	–11.6, –11.5						–11.52
I_b^3	KOW	–7.40, –7.30	14370	11596	35.5	48	25	–7.38
	LP	–2.90, –2.80						–2.81
	SL	–11.6, –11.4						–11.52
I_b^4	KOW	–7.50, –7.40	14166	11588	71.7	49	26	–7.48
	LP	–0.70, –0.60						–0.63
	SL	–11.4, –11.2						–11.39

6. Conclusions

In this paper, we studied the problem of inferring chemical structures from desired chemical properties and constraints, based on the framework proposed and developed in [18–20]. In the previous applications of the framework of inferring chemical graphs, artificial neural network (ANN) and decision tree have been used for the machine learning of Stage 3. In this paper, we used linear regression in Stage 3 for the first time and derived an MILP formulation that simulates the computation process of linear regression. We also extended a way of specifying a target value y^* in a property so that the predicted value $\eta(f(\mathbb{C}^\dagger))$ of a target chemical graph $\mathbb{C}^\dagger$ is required to belong to an interval between two specified values $\underline{y}^*$ and $\overline{y}^*$. Furthermore, we modified a model of chemical compounds so that multivalence chemical elements, cation and anion are treated, and introduced the rank and the adjacency-configuration of leaf-edges as new descriptors in a feature vector of a chemical graph.

We implemented the new system of the framework and conducted computational experiments for Stages 1 to 5. We found 18 properties for which linear regression delivers a relatively good prediction function by using our feature vector based on the two-layered model. We also observed that an MILP formulation for inferring a chemical graph in Stage 4 can be solved efficiently over different types of test instances with complicated topological specifications. The experimental result suggests that our method can infer chemical graphs with up to 50 non-hydrogen atoms. Therefore, combination of linear regression and integer programming is a potentially useful approach to computational molecular design.

It is an interesting future work to use other learning methods such as graph convolution networks, random forest and an ensemble method to construct a prediction function and derive the corresponding MILP formulations in Stages 3 and 4 in the framework.

Author Contributions

Conceptialization, HN and TA; methodology, HN; software, JZ, NAA and KH; validation, JZ, NAA and HN; formal analysis, HN; data resources, KH, LZ, HN and TA; writing—original draft preparation, HN; writing—review and editing, NAA and TA; project administration, HN; funding acquisition, TA. All authors contributed to editorial changes in the manuscript. All authors read and approved the final manuscript.

Ethics Approval and Consent to Participate

Not applicable.

Acknowledgment

Not applicable.

Funding

This research was supported, in part, by Japan Society for the Promotion of Science, Japan, under Grant #18H04113.

Conflict of Interest

The authors declare no conflict of interest. TA is serving as the guest editor of this journal. We declare that TA had no involvement in the peer review of this article and has no access to information regarding its peer review. Full responsibility for the editorial process for this article was delegated to AK and GP.

Supplementary Material

Supplementary material associated with this article can be found, in the online version, at https://doi.org/10.31083/j.fbl2706188.

References

[1] Cherkasov A, Muratov EN, Fourches D, Varnek A, Baskin II, Cronin M, *et al*. QSAR modeling: where have you been? Where are you going to? Journal of Medicinal Chemistry. 2014; 57: 4977–5010.

[2] Lo YC, Rensi SE, Torng W, Altman RB. Machine learning in

chemoinformatics and drug discovery. Drug Discovery Today. 2018; 23: 1538–1546.
[3] Tetko IV, Engkvist O. From Big Data to Artificial Intelligence: chemoinformatics meets new challenges. Journal of Cheminformatics. 2020; 12: 74.
[4] Ghasemi F, Mehridehnavi A, Pérez-Garrido A, Pérez-Sánchez H. Neural network and deep-learning algorithms used in QSAR studies: merits and drawbacks. Drug Discovery Today. 2018; 23: 1784–1790.
[5] Miyao T, Kaneko H, Funatsu K. Inverse QSPR/QSAR Analysis for Chemical Structure Generation (from y to x). Journal of Chemical Information and Modeling. 2016; 56: 286–99.
[6] Ikebata H, Hongo K, Isomura T, Maezono R, Yoshida R. Bayesian molecular design with a chemical language model. The Journal of Computer-Aided Molecular Design. 2017; 31: 379–391.
[7] Rupakheti C, Virshup A, Yang W, Beratan DN. Strategy to discover diverse optimal molecules in the small molecule universe. Journal of Chemical Information and Modeling. 2015; 55: 529-5-37.
[8] Bohacek RS, McMartin C, Guida WC. The art and practice of structure-based drug design: a molecular modeling perspective. Medicinal Research Reviews. 1996; 16: 3–50.
[9] Akutsu T, Fukagawa D, Jansson J, Sadakane K. Inferring a graph from path frequency. Discrete Applied Mathematics. 2012; 160: 1416–1428.
[10] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv. 2016. (in press)
[11] Segler MHS, Kogej T, Tyrchan C, Waller MP. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. ACS Central Science. 2018; 4: 120–131.
[12] Yang X, Zhang J, Yoshizoe K, Terayama K, Tsuda K. ChemTS: an efficient python library for de novo molecular generation. Science and Technology of Advanced Materials. 2017; 18: 972–976.
[13] Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, *et al*. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. ACS Central Science. 2018; 4: 268–276.
[14] Kusner MJ, Paige B, Hernández-Lobato JM. Grammar variational autoencoder. Proceedings of the 34th International Conference on Machine Learning. 2017; 70: 1945–1954.
[15] Madhawa K, Ishiguro K, Nakago K, Abe M. GraphNVP: an invertible flow model for generating molecular graphs. arXiv. 2019. (in press)
[16] Shi C, Xu M, Zhu Z, Zhang W, Zhang M, Tang J. GraphAF: a flow-based autoregressive model for molecular graph generation. arXiv. 2020. (in press)
[17] De Cao N, Kipf T. MolGAN: An implicit generative model for small molecular graphs. arXiv. 2018. (in press)
[18] Akutsu T, Nagamochi H. A mixed integer linear programming formulation to artificial neural networks. Proceedings of the 2019 2nd International Conference on Information Science and Systems. 2019; 215–220.
[19] Azam NA, Chiewvanichakorn R, Zhang F, Shurbevski A, Nagamochi H, Akutsu T. A method for the inverse QSAR/QSPR based on artificial neural networks and mixed integer linear programming. Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies. 2020; 3: 101–108.
[20] Zhang F, Zhu J, Chiewvanichakorn R, Shurbevski A, Nagamochi H, Akutsu T. 'A new integer linear programming formulation to the inverse QSAR/QSPR for acyclic chemical compounds using skeleton trees'. The 33rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Kitakyushu, Japan. 2020.
[21] Azam NA, Zhu J, Sun Y, Shi Y, Shurbevski A, Zhao L, *et al*. A novel method for inference of acyclic chemical compounds with bounded branch-height based on artificial neural networks and integer programming. Algorithms for Molecular Biology. 2021; 16: 18.
[22] Ito R, Azam NA, Wang C, Shurbevski A, Nagamochi H, Akutsu T. 'A novel method for the inverse QSAR/QSPR to monocyclic chemical compounds based on artificial neural networks and integer programming'. BIOCOMP2020. Las Vegas, Nevada, USA. 2020.
[23] Zhu J, Wang C, Shurbevski A, Nagamochi H, Akutsu T. A novel method for inference of chemical compounds of cycle index two with desired properties based on artificial neural networks and integer programming. Algorithms. 2020; 13: 124.
[24] Akutsu T, Nagamochi H. A novel method for inference of chemical compounds with prescribed topological substructures based on integer programming. arXiv. 2020. (in press)
[25] Tanaka K, Zhu J, Azam NA, Haraguchi K, Zhao L, Nagamochi H, Akutsu T. 'An inverse QSAR method based on decision tree and integer programming'. The 17th International Conference on Intelligent Computing. Shenzhen, China. 2021.
[26] Shi Y, Zhu J, Azam NA, Haraguchi K, Zhao L, Nagamochi H, *et al*. An Inverse QSAR Method Based on a Two-Layered Model and Integer Programming. International Journal of Molecular Sciences. 2021; 22: 2847.
[27] Zhu J, Azam NA, Zhang F, Shurbevski A, Haraguchi K, Zhao L, *et al*. A Novel Method for Inferring Chemical Compounds with Prescribed Topological Substructures Based on Integer Programming. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2021. (in press)
[28] Hoerl A, Kennard R. Ridge regression. In Encyclopedia of Statistical Sciences (pp. 129–136). New York: Wiley. 1988.
[29] Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological). 1996; 58: 267–288.
[30] Annotations from HSDB (on pubchem). Available at: https://pubchem.ncbi.nlm.nih.gov/ (Accessed: 16 February 2022).
[31] Jalali-Heravi M, Fatemi MH. Artificial neural network modeling of Kovats retention indices for noncyclic and monocyclic terpenes. Journal of Chromatography A. 2001; 915: 177–183.
[32] Roy K, Saha A. Comparative QSPR studies with molecular connectivity, molecular negentropy and TAU indices. Journal of Molecular Modeling. 2003; 9: 259–270.
[33] MoleculeNet. Available at: https://moleculenet.org (Accessed: 16 February 2022).
[34] Goussard V, François Duprat F, Ploix J-L, Dreyfus G, Nardello-Rataj V, Aubry J-M. A new machine-learning tool for fast estimation of liquid viscosity. application to cosmetic oils. Journal of Chemical Information and Modeling. 2020; 60: 2012–2023.
[35] Naef R. 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. 2019; 24: 1626.
[36] Wang JB, Cao DS, Zhu MF, Yun YH, Xiao N, Liang YZ. In silico evaluation of logD7.4 and comparison with other prediction methods. Journal of Chemometrics. 2015; 29: 389–398.