A novel method for inference of chemical compounds of cycle index two with desired properties based on artificial neural networks and integer programming

Zhu, Jianshen; Wang, Chenxi; Shurbevski, Aleksandar; Nagamochi, Hiroshi; Akutsu, Tatsuya

algorithms

MDPI

Article

A Novel Method for Inference of Chemical Compounds of Cycle Index Two with Desired Properties Based on Artificial Neural Networks and Integer Programming

Jianshen Zhu [1,†], Chenxi Wang [1,†], Aleksandar Shurbevski [1], Hiroshi Nagamochi [1,*] and Tatsuya Akutsu [2,*]

1 Department of Applied Mathematics and Physics, Kyoto University, Kyoto 606-8501, Japan; zhujs@amp.i.kyoto-u.ac.jp (J.Z.); chenxi@amp.i.kyoto-u.ac.jp (C.W.); shurbevski@amp.i.kyoto-u.ac.jp (A.S.)
2 Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji 611-0011, Japan
* Correspondence: nag@amp.i.kyoto-u.ac.jp (H.N.); takutsu@kuicr.kyoto-u.ac.jp (T.A.); Tel.: +81-75-753-4920 (H.N.); +81-774-38-3015 (T.A.)
† These authors contributed equally to this work.

Received: 22 April 2020; Accepted: 13 May 2020; Published: 18 May 2020

Abstract: Inference of chemical compounds with desired properties is important for drug design, chemo-informatics, and bioinformatics, to which various algorithmic and machine learning techniques have been applied. Recently, a novel method has been proposed for this inference problem using both artificial neural networks (ANN) and mixed integer linear programming (MILP). This method consists of the training phase and the inverse prediction phase. In the training phase, an ANN is trained so that the output of the ANN takes a value nearly equal to a given chemical property for each sample. In the inverse prediction phase, a chemical structure is inferred using MILP and enumeration so that the structure can have a desired output value for the trained ANN. However, the framework has been applied only to the case of acyclic and monocyclic chemical compounds so far. In this paper, we significantly extend the framework and present a new method for the inference problem for rank-2 chemical compounds (chemical graphs with cycle index 2). The results of computational experiments using such chemical properties as octanol/water partition coefficient, melting point, and boiling point suggest that the proposed method is much more useful than the previous method.

Keywords: mixed integer linear programming; QSAR/QSPR; molecular design

1. Introduction

Inference of chemical compounds with desired properties is important for computer-aided drug design. Since drug design is one of the major targets of chemo-informatics and bioinformatics, it is also important in these areas. Indeed, this problem has been extensively studied in chemo-informatics under the name of inverse QSAR/QSPR [1,2], where QSAR/QSPR denotes Quantitative Structure Activity/Property Relationships. Since chemical compounds are usually represented as undirected graphs, this problem is important also from graph theoretic and algorithmic viewpoints.

Inverse QSAR/QSPR is often formulated as an optimization problem to find a chemical graph maximizing (or minimizing) an objective function under various constraints, where objective functions reflect certain chemical activities or properties. In many cases, objective functions are derived from a set of training data consisting of known molecules and their activities/properties using statistical and machine learning methods.

In both forward and inverse QSAR/QSPR, chemical graphs are often represented as vectors of real or integer numbers because it is difficult to directly handle graphs using statistical and machine learning methods. Elements of these vectors are called *descriptors* in QSAR/QSPR studies, and these vectors correspond to *feature vectors* in machine learning. Using these chemical descriptors, various heuristic and statistical methods have been developed for finding optimal or nearly optimal graph structures under given objective functions [1,3,4]. In many cases, inference or enumeration of graph structures from a given feature vector is a crucial subtask in these methods. Various methods have been developed for this enumeration problem [5–8] and the computational complexity of the inference problem has been analyzed [9–11]. On the other hand, enumeration in itself is a challenging task, since the number of molecules (i.e., chemical graphs) with up to 30 atoms (vertices) `C`, `N`, `O`, and `S`, may exceed 10^{60} [12].

As in many other fields, Artificial Neural Network (ANN) and deep learning technologies have recently been applied to inverse QSAR/QSPR. For example, variational autoencoders [13], recurrent neural networks [14,15], and grammar variational autoencoders [16] have been applied. In these approaches, new chemical graphs are generated by solving a kind of inverse problems on neural networks, where neural networks are trained using known chemical compound/activity pairs. However, the optimality of the solution is not necessarily guaranteed in these approaches. In order to guarantee the optimality, a novel approach has been proposed [17] for ANNs with ReLU activation functions and sigmoid activation functions, using mixed integer linear programming (MILP). In their approach, activation functions on neurons are efficiently encoded as piece-wise linear functions so as to represent ReLU functions exactly and sigmoid functions approximately.

Recently, a new framework has been proposed [18–20] by combining two previous approaches; efficient enumeration of tree-like graphs [5], and MILP-based formulation of the inverse problem on ANNs [17]. This combined framework for inverse QSAR/QSPR mainly consists of two phases, one for constructing a prediction function to a chemical property, and the other for constructing graphs based on the inverse of the prediction function. The first phase solves (I) PREDICTION PROBLEM, where a prediction function $\psi_{\mathcal{N}}$ on a chemical property π is constructed with an ANN $\mathcal{N}$ using a data set of chemical compounds G and their values $a(G)$ of π. The second phase solves (II) INVERSE PROBLEM, where (II-a) given a target value y^* of the chemical property π, a feature vector x^* is inferred from the trained ANN $\mathcal{N}$ so that $\psi_{\mathcal{N}}(x^*)$ is close to y^* and (II-b) then a set of chemical structures G^* such that $f(G^*) = x^*$ is enumerated. In (II-b) of the above-mentioned previous methods [18–20], an MILP is formulated for acyclic chemical compounds. Their methods were applicable only to acyclic chemical graphs (i.e., tree-structured chemical graphs), where the ratio of acyclic chemical graphs in a major chemical database (PubChem) is 2.91%. Afterward, Ito et al. [21] designed a method of inferring monocyclic chemical graphs (chemical graphs with cycle index or rank 1) by formulating a new MILP and using an efficient algorithm for enumerating monocyclic chemical graphs [22]. This still leaves a big limitation because the ratio of acyclic and monocyclic chemical graphs in the chemical database PubChem is only 16.26%.

To break this limitation, we significantly extend the MILP-based approach for inverse QSAR/QSPR so that "rank-2 chemical compounds" (chemical graphs with cycle index or rank 2) can be efficiently handled, where the ratio of chemical graphs with rank at most 2 in the database PubChem is 44.5%. Note that there are three different topological structures, called *polymer-topologies* over all rank-2 chemical compounds. In particular, we propose a novel MILP formulation for (II-a) along with a new set of descriptors. One big advantage of this new formulation is that an MILP instance has a solution if and only if there exists a rank-2 chemical graph satisfying given constraints, which is useful to significantly reduce redundant search in (II-b). We conducted computational experiments to infer rank-2 chemical compounds on several chemical properties.

The paper is organized as follows. Section 2.1 introduces some notions on graphs, a modeling of chemical compounds, and a choice of descriptors. Section 2.2 reviews the framework for inferring chemical compounds based on ANNs and MILPs. Section 2.3 introduces a method of modeling rank-2 chemical graphs with different cyclic structures in a unified way and proposes an MILP formulation

that represents a rank-2 chemical graph G of n vertices, where our MILP requires only $O(n)$ variables and constraints when the maximum height of subtrees in G is constant. Section 3 reports the results on some computational experiments conducted for chemical properties such as octanol/water partition coefficient, melting point, and boiling point. Section 4 makes some concluding remarks. Appendix A provides the detail of all variables and constraints in our MILP formulation.

2. Materials and Methods

2.1. Preliminary

This section introduces some notions and terminology on graphs, a modeling of chemical compounds, and our choice of descriptors.

2.1.1. Multigraphs and Graphs

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

Multigraphs

A *multigraph* is defined to be a pair (V, E) of a vertex set V and an edge set E such that each edge $e \in E$ joins two vertices $u, v \in V$ (possibly $u = v$) and the vertices u and v are called the *end-vertices* of the edge e, and let $V(e)$ denote the set of the end-vertices of an edge $e \in E$, where an edge e with $|V(e)| = 1$ is called a *loop*. We denote the vertex and edge sets of a multigraph M by $V(M)$ and $E(M)$, respectively. A path with end-vertices u and v is called a *u, v-path*, and the length of a path is defined to be the number of edges in the path.

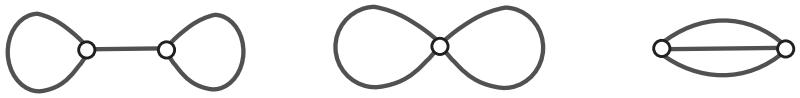
Let M be a multigraph. An edge $e \in E(M)$ is called *multiple* (to an edge $e' \in E(M)$) if there is another edge $e' \in E(M)$ with $V(e) = V(e')$. For a vertex $v \in V(M)$, the set of neighbors of v in M is denoted by $N_M(v)$, and the *degree* $\deg_M(v)$ of v is defined to be the number of times an edge in $E(M)$ is incident to v; i.e., $\deg_M(v) = |\{e \in E(M) \mid v \in V(e), |V(e)| = 2\}| + 2|\{e \in E(M) \mid v \in V(e), |V(e)| = 1\}|$. A multigraph is called *simple* if it has no loop and there is at most one edge between any two vertices. We observe that the sum of the degrees over all vertices is twice the number of edges in any multigraph M; i.e.,

$$2|E(M)| = \sum_{v \in V(M)} \deg_M(v).$$

For a subset X of vertices in M, let $M - X$ denote the multigraph obtained from M by removing the vertices in X and any edge incident to a vertex in X. An operation of *subdividing* a non-loop edge (resp., loop) $e \in E(M)$ with $V(e) = \{v_1, v_2\}$ (resp., $V(e) = \{v_1 = v_2\}$) is to replace e with two new edges e_1 and e_2 incident to a new vertex v_e such that each e_i is incident to v_i. An operation of *contracting* a vertex u of degree 2 in M is to replace the two edges uv and uv' incident to u with a single edge vv' removing vertex u, where the resulting edge is a loop when $v = v'$. The *rank* $\mathrm{r}(M)$ of a multigraph M is defined to be the minimum number of edges to be removed to make the multigraph acyclic. We call a multigraph M with $\mathrm{r}(M) = k$ a *rank-k graph*. Let $V_{\deg,i}(M)$ denote the set of vertices of degree i in M. The *core* $\mathrm{Cr}(M)$ of M is defined to be an induced subgraph M^* that is obtained from $M' := M$ by setting $M' := M' - V_{\deg,1}(M')$ repeatedly until M^* contains at most two vertices or consists of vertices of degree at least 2. The core M^* of a connected multigraph M consists of a single vertex (resp., two vertices) if and only if M is a tree with an even (resp., odd) diameter. A vertex (resp., an edge) in M is called a *core vertex* (resp., *core edge*) if it is contained in the core of M and is called a *non-core vertex* (resp., *non-core edge*) otherwise. The *core size* $\mathrm{cs}(M)$ is defined to be the number of core vertices of M, and the *core height* $\mathrm{ch}(M)$ is defined to be the maximum length of a path between a vertex $v \in V(M^*)$ to a leaf of M without passing through any core edge. The set of non-core edges induces a collection of subtrees, each of which we call a *non-core component* of M, where each non-core component C contains

exactly one core vertex v_C and we regard C as a tree rooted at v_C. Let C be a non-core component of M. The *height* height(v) of a vertex v in C is defined to be the maximum length of a path from v to a leaf u in the descendants of v.

A multigraph is called a *polymer topology* if it is connected and the degree of every vertex is at least 3. Tezuka and Oike [23] pointed out that a classification of polymer topologies will lay a foundation for elucidation of structural relationships between different macro-chemical molecules and their synthetic pathways. For integers $r \geq 0$ and $d \geq 3$, let $\mathcal{PT}(r,d)$ denote the set of all rank-r polymer topologies with maximum degree at most d. Figure 1 illustrates the three rank-2 polymer topologies in $\mathcal{PT}(2,4)$.

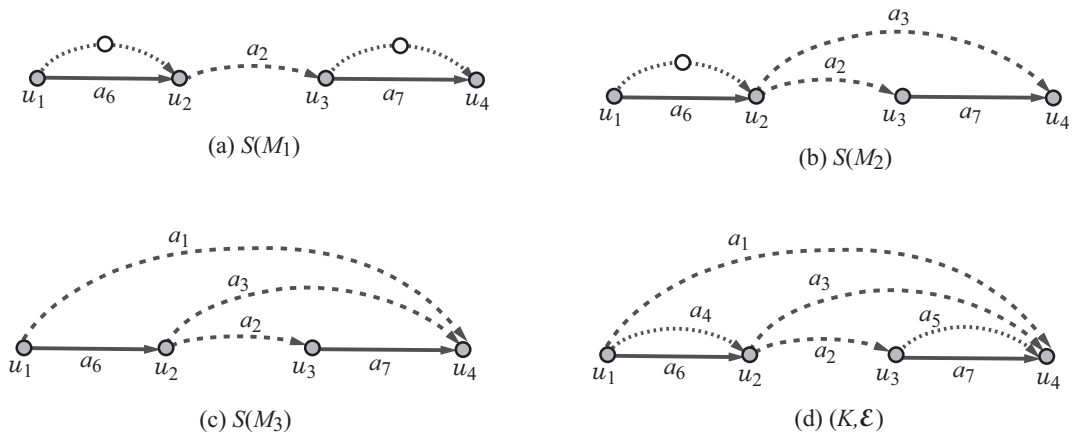


(a) M_1 (b) M_2 (c) M_3

Figure 1. An illustration of the three rank-2 polymer topologies $M_1, M_2, M_3 \in \mathcal{PT}(2,4)$.

For a polymer topology M, the *least simple graph* $S(M)$ of M is defined to be a simple graph obtained from M by subdividing each loop in M with two new vertices of degree 2 and subdividing all multiple edges (except for one) between every two adjacent vertices in M. Note that $|V(S(M))| = |V(M)| + r + s$ for the rank r of M and the number s of loops in M.

The *polymer topology* $\mathrm{Pt}(M)$ of a multigraph M with $\mathrm{r}(M) \geq 2$ is defined to be a multigraph M' of degree at least 3 that is obtained from the core $\mathrm{Cr}(M)$ by contracting all vertices of degree 2. Note that $\mathrm{r}(\mathrm{Pt}(M)) = \mathrm{r}(M)$. Figure 2a–c illustrate the least simple graph $S(M)$ of each polymer topology $M \in \mathcal{PT}(2,4)$, where Figure 2d illustrates a graph that contains all least simple graphs.



(a) $S(M_1)$ (b) $S(M_2)$

(c) $S(M_3)$ (d) $(K,\mathcal{E})$

Figure 2. An illustration of the least simple graphs of the rank-2 polymer topologies $M_1, M_2, M_3 \in \mathcal{PT}(2,4)$ in Figure 1 and a scheme graph $(K,\mathcal{E})$: **(a)** $S(M_1)$; **(b)** $S(M_2)$; **(c)** $S(M_3)$; **(d)** a scheme graph $(K = (\{u_1,u_2,u_3,u_4\},E), \mathcal{E} = (E_1,E_2,E_3))$ where each edge u_iu_j is directed from one end-vertex u_i to the other end-vertex u_j with $i < j$, and $E_1 = \{a_1 = (u_1,u_4), a_2 = (u_2,u_3), a_3 = (u_2,u_4)\}$, $E_2 = \{a_4 = (u_1,u_2), a_5 = (u_3,u_4)\}$ and $E_3 = \{a_6 = (u_1,u_2), a_7 = (u_3,u_4)\}$, and the edges in E_1 (resp., E_2 and E_3) are depicted with dashed (resp., dotted and solid) lines.

Graphs

Let $H = (V,E)$ be a graph with a set V of vertices and a set E of edges. Define the *1-path connectivity* $\kappa_1(H)$ of H to be $\sum_{uv \in E} 1/\sqrt{\deg_H(u)\deg_H(v)}$.

Let H be a rank-2 connected graph such that the maximum degree is at most 4. We see that H contains two vertices v_a and v_b such that either there are three disjoint paths between v_a and v_b or H contains two edge disjoint cycles C and C', which are joined with a path between v_a and v_b

(possibly $v_a = v_b$). We introduce the *topological parameter* $\theta(H)$ of rank-2 connected graph H as follows. When H has three disjoint paths between v_a and v_b, define $\theta(H)$ to be the minimum number of edges along a path between v_a and v_b. When H contains two edge disjoint cycles C and C', which are joined with a path P between v_a and v_b (possibly $v_a = v_b$), define $\theta(H)$ to be $-|E(P)|$.

For positive integers a, b and c with $b \geq 2$, let $T(a, b, c)$ denote the rooted tree such that the number of children of the root is a, the number of children of each non-root internal vertex is b and the distance from the root to each leaf is c. In the rooted tree $T(a, b, c)$, we denote the vertices by $v_1, v_2, \ldots, v_n$ ($n = a(b^c - 1)/(b-1) + 1$) with a breadth-first-search order, and denote the edge between a vertex v_i with $i \in [2, n]$ and its parent by e_i. For each vertex v_i in $T(a, b, c)$, let $\mathrm{Cld}(i)$ denote the set of indices j such that v_j is a child of v_i, and $\mathrm{prt}(i)$ denote the index j such that v_j is the parent of v_i when $i \in [2, n]$.

2.1.2. Modeling of Chemical Compounds

Chemical Graphs

We represent the graph structure of a chemical compound as a graph with labels on vertices and multiplicity on edges in a hydrogen-suppressed model. Nearly 68.5% (resp., 99%) of the rank-2 chemical graphs with at most 200 non-hydrogen atoms registered in chemical database PubChem have a maximum degree at most 3 (resp., 4) for all non-core vertices in the hydrogen-suppressed model.

Let Λ be a set of labels, each of which represents a chemical element such as `C` (carbon), `O` (oxygen), `N` (nitrogen) and so on, where we assume that Λ does not contain `H` (hydrogen). Let $\mathrm{mass}(\mathtt{a})$ and $\mathrm{val}(\mathtt{a})$ denote the mass and valence of a chemical element $\mathtt{a} \in \Lambda$, respectively. In our model, we use integers $\mathrm{mass}^*(\mathtt{a}) = \lfloor 10 \cdot \mathrm{mass}(\mathtt{a}) \rfloor$, $\mathtt{a} \in \Lambda$ and assume that each chemical element $\mathtt{a} \in \Lambda$ has a unique valence $\mathrm{val}(\mathtt{a}) \in [1, 4]$.

We introduce a total order $<$ over the elements in Λ according to their mass values; i.e., we write $\mathtt{a} < \mathtt{b}$ for chemical elements $\mathtt{a}, \mathtt{b} \in \Lambda$ with $\mathrm{mass}(\mathtt{a}) < \mathrm{mass}(\mathtt{b})$. Choose a set $\Gamma_<$ of tuples $\gamma = (\mathtt{a}, \mathtt{b}, k) \in \Lambda \times \Lambda \times [1, 3]$ such that $\mathtt{a} < \mathtt{b}$. For a tuple $\gamma = (\mathtt{a}, \mathtt{b}, k) \in \Lambda \times \Lambda \times [1, 3]$, let $\overline{\gamma}$ denote the tuple $(\mathtt{b}, \mathtt{a}, k)$. Set $\Gamma_> = \{\overline{\gamma} \mid \gamma \in \Gamma_<\}$, $\Gamma_= = \{(\mathtt{a}, \mathtt{a}, k) \mid \mathtt{a} \in \Lambda, k \in [1, 3]\}$ and $\Gamma = \Gamma_< \cup \Gamma_=$. A pair of two atoms $\mathtt{a}$ and $\mathtt{b}$ joined with a bond of multiplicity k is denoted by a tuple $\gamma = (\mathtt{a}, \mathtt{b}, k) \in \Gamma$, called the *adjacency-configuration* of the atom pair.

We use a hydrogen-suppressed model because hydrogen atoms can be added at the final stage. A *chemical graph* over Λ and Γ is defined to be a tuple $G = (H, \alpha, \beta)$ of a graph $H = (V, E)$, a function $\alpha : V \to \Lambda$ and a function $\beta : E \to [1, 3]$ such that

(i) H is connected;
(ii) $\sum_{uv \in E} \beta(uv) \leq \mathrm{val}(\alpha(u))$ for each vertex $u \in V$; and
(iii) $(\alpha(u), \alpha(v), \beta(uv)) \in \Gamma$ for each edge $uv \in E$.

Let $\mathcal{G}(\Lambda, \Gamma)$ denote the set of chemical graphs over Λ and Γ.

Descriptors

In our method, we use only graph-theoretical descriptors for defining a feature vector, which facilitates our designing an algorithm for constructing graphs. Given a chemical graph $G = (H, \alpha, \beta)$, we define a *feature vector* $f(G)$ that consists of the following 14 kinds of descriptors:

- $n(G)$: the number of vertices in G;
- $\mathrm{cs}(G)$: the core size of G;
- $\mathrm{ch}(G)$: the core height of G;
- $\kappa_1(G)$: the 1-path connectivity of G;
- $\mathrm{dg}_i(G)$ ($i \in [1, 4]$): the number of vertices of degree i in G;
- $\mathrm{ce}^{\mathrm{co}}_{\mathtt{a}}(G)$ ($\mathtt{a} \in \Lambda$): the number of core vertices with chemical element $\mathtt{a} \in \Lambda$;
- $\mathrm{ce}^{\mathrm{nc}}_{\mathtt{a}}(G)$ ($\mathtt{a} \in \Lambda$): the number of non-core vertices with chemical element $\mathtt{a} \in \Lambda$;

- $\overline{\mathrm{ms}}(G)$: the average of mass* of atoms in G;
- $b_k^{\mathrm{co}}(G)$ ($k \in [2,3]$): the number of double and triple bonds in core edges;
- $b_k^{\mathrm{nc}}(G)$ ($k \in [2,3]$): the number of double and triple bonds in non-core edges;
- $\mathrm{ac}_\gamma^{\mathrm{co}}(G)$ ($\gamma = (\mathtt{a},\mathtt{b},k) \in \Gamma$): the number of adjacency-configurations $(\mathtt{a},\mathtt{b},k)$ of core edges;
- $\mathrm{ac}_\gamma^{\mathrm{nc}}(G)$ ($\gamma = (\mathtt{a},\mathtt{b},k) \in \Gamma$): the number of adjacency-configurations $(\mathtt{a},\mathtt{b},k)$ of non-core edges;
- $\theta(H)$: the topological parameter of H; and
- $n_{\mathtt{H}}(G)$: the number of hydrogen atoms to be included in G; i.e.,
  $n_{\mathtt{H}}(G) = \sum_{\mathtt{a}\in\Lambda} \mathrm{val}(\mathtt{a})(\mathrm{ce}_{\mathtt{a}}^{\mathrm{co}}(G) + \mathrm{ce}_{\mathtt{a}}^{\mathrm{nc}}(G)) - 2(n(G) + 1 + b_2^{\mathrm{co}}(G) + b_2^{\mathrm{nc}}(G) + 2b_3^{\mathrm{co}}(G) + 2b_3^{\mathrm{nc}}(G))$.

The number k of descriptors in our feature vector $x = f(G)$ is $k = 2|\Lambda| + 2|\Gamma| + 15$.

2.2. A Method for Inferring Chemical Graphs

This section reviews the framework that solves the inverse QSAR/QSPR by using MILPs [18]. For a specified chemical property π such as boiling point, we denote by $a(G)$ the observed value of the property π for a chemical compound G. As the Phase 1, we solve (I) PREDICTION PROBLEM with the following three steps.

Phase 1.

Step 1: Let DB be a set of chemical graphs. For a specified chemical property π, choose a class $\mathcal{G}$ of graphs such as acyclic graphs or monocyclic graphs. Prepare a data set $D_\pi = \{G_i \mid i = 1,2,\ldots,m\} \subseteq \mathcal{G} \cap \mathrm{DB}$ such that the value $a(G_i)$ of each chemical graph G_i, $i = 1,2,\ldots,m$ is available. Set reals $\underline{a}, \overline{a} \in \mathbb{R}$ so that $\underline{a} \le a(G_i) \le \overline{a}$, $i = 1,2,\ldots,m$. See Figure 3 for an illustration of Step 1.

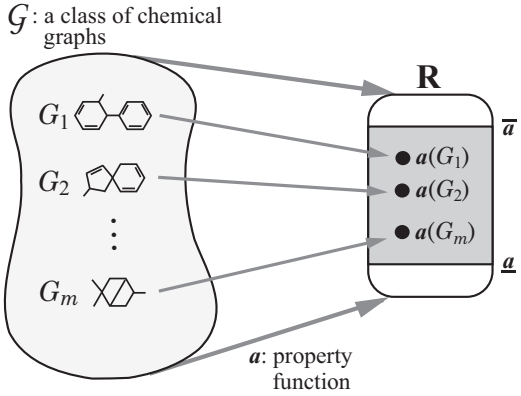


Figure 3. An illustration of Step 1: A data set D_π of chemical graphs G_i, $i = 1,2,\ldots,m$ in a class $\mathcal{G}$ of graphs whose values $a(G_i) \in [\underline{a}, \overline{a}]$ of a chemical property π are available.

Step 2: Introduce a feature function $f : \mathcal{G} \to \mathbb{R}^k$ for a positive integer k. We call $f(G)$ the *feature vector* of $G \in \mathcal{G}$, and call each entry of a vector $f(G)$ a *descriptor* of G. See Figure 4 for an illustration of Step 2.

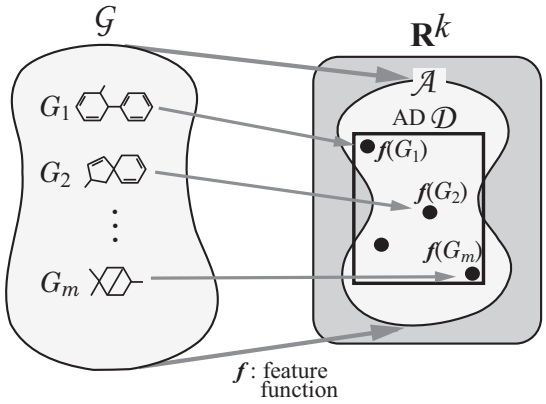


Figure 4. An illustration of Step 2: Each chemical graph $G \in \mathcal{G}$ is mapped to a vector $f(G)$ in a feature vector space $\mathbb{R}^k$ for some positive integer k.

Step 3: Construct a prediction function $\psi_{\mathcal{N}}$ with an ANN $\mathcal{N}$ that, given a vector in $\mathbb{R}^k$, returns a real in the range $[\underline{a}, \overline{a}]$ so that $\psi_{\mathcal{N}}(f(G))$ takes a value nearly equal to $a(G)$ for many chemical graphs in D. See Figure 5 for an illustration of Step 3.

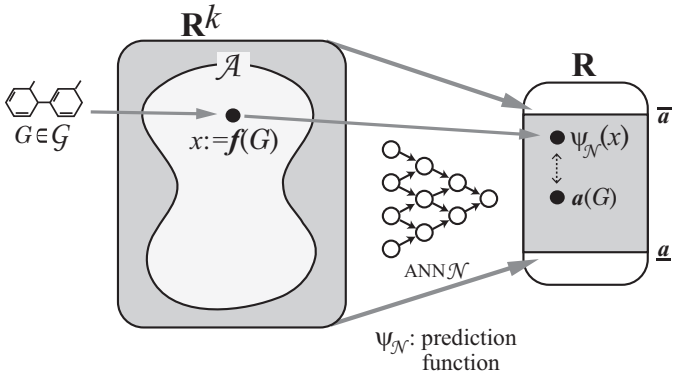


Figure 5. An illustration of Step 3: A prediction function $\psi_{\mathcal{N}}$ from the feature vector space $\mathbb{R}^k$ to the range $[\underline{a}, \overline{a}]$ is constructed based on an ANN $\mathcal{N}$.

Next we explain how to solve the inverse problem to the prediction in Phase 1 using an MILP formulation. A vector $x \in \mathbb{R}^k$ is called *admissible* if there is a graph $G \in \mathcal{G}$ such that $f(G) = x$ [18]. Let $\mathcal{A}$ denote the set of admissible vectors $x \in \mathbb{R}^k$. In this paper, we use the range-based method to define an applicability domain (AD) [24] to our inverse QSAR/QSPR. Set $\underline{x_j}$ and $\overline{x_j}$ to be the minimum and maximum values of the j-th descriptor x_j in $f(G_i)$ over all graphs G_i, $i = 1, 2, \ldots, m$ (where we possibly normalize some descriptors such as $\mathrm{ce}^{\mathrm{co}}_{\mathtt{a}}(G)$, which is normalized with $\mathrm{ce}^{\mathrm{co}}_{\mathtt{a}}(G)/n(H)$). Define our AD $\mathcal{D}$ to be the set of vectors $x \in \mathbb{R}^k$ such that $\underline{x_j} \leq x_j \leq \overline{x_j}$ for the variable x_j of each j-th descriptor, $j = 1, 2, \ldots, k$. As the second phase, we solve (II) Inverse Problem for the inverse QSAR/QSPR by treating the following inference problems.

(II-a) Inference of Vectors
Input: A real $y^* \in [\underline{a}, \overline{a}]$.
Output: Vectors $x^* \in \mathcal{A} \cap \mathcal{D}$ and $g^* \in \mathbb{R}^h$ such that $\psi_{\mathcal{N}}(x^*) = y^*$ and g^* forms a chemical graph $G^* \in \mathcal{G}$ with $f(G^*) = x^*$.

(II-b) Inference of Graphs
Input: A vector $x^* \in \mathcal{A} \cap \mathcal{D}$.
Output: All graphs $G^* \in \mathcal{G}$ such that $f(G^*) = x^*$.

To treat Problem (II-a), we use MILPs for inferring vectors in ANNs [17]. In MILPs, we can easily impose additional linear constraints or fix some variables to specified constants. We include into the MILP a linear constraint such that $x \in \mathcal{D}$ to obtain the next result.

Theorem 1. *Let $\mathcal{N}$ be an ANN with a piecewise-linear activation function for an input vector $x \in \mathbb{R}^k$, n_A denote the number of nodes in the architecture and n_B denote the total number of break-points over all activation functions. Then there is an MILP $\mathcal{M}(x, y; \mathcal{C}_1)$ that consists of variable vectors $x \in \mathcal{D}$ $(\subseteq \mathbb{R}^k)$, $y \in \mathbb{R}$, and an auxiliary variable vector $z \in \mathbb{R}^p$ for some integer $p = O(n_A + n_B)$ and a set $\mathcal{C}_1$ of $O(n_A + n_B)$ constraints on these variables such that: $\psi_{\mathcal{N}}(x^*) = y^*$ if and only if there is a vector (x^*, y^*) feasible to $\mathcal{M}(x, y; \mathcal{C}_1)$.*

See Appendix A for the set of constraints to define our AD $\mathcal{D}$ in the MILP $\mathcal{M}(x, y; \mathcal{C}_1)$ in Theorem 1.

To attain the admissibility of inferred vector x^*, we also introduce a variable vector $g \in \mathbb{R}^q$ for some integer q and a set $\mathcal{C}_2$ of constraints on x and g such that $x^* \in \mathcal{A}$ holds in the following sense: (x^*, g^*) is feasible to the MILP $\mathcal{M}(x, g; \mathcal{C}_2)$ if and only if g^* forms a chemical graph $G^* \in \mathcal{G}$ with $f(G^*) = x^*$. The Phase 2 consists of the next two steps.

Phase 2.

Step 4: Formulate Problem (II-a) as the above MILP $\mathcal{M}(x, y, g; \mathcal{C}_1, \mathcal{C}_2)$ based on $\mathcal{G}$ and $\mathcal{N}$. Find a set F^* of vectors $x^* \in \mathcal{A} \cap \mathcal{D}$ such that $(1 - \varepsilon)y^* \leq \psi_{\mathcal{N}}(x^*) \leq (1 + \varepsilon)y^*$ for a tolerance ε set to be a small positive real. See Figure 6 for an illustration of Step 4.

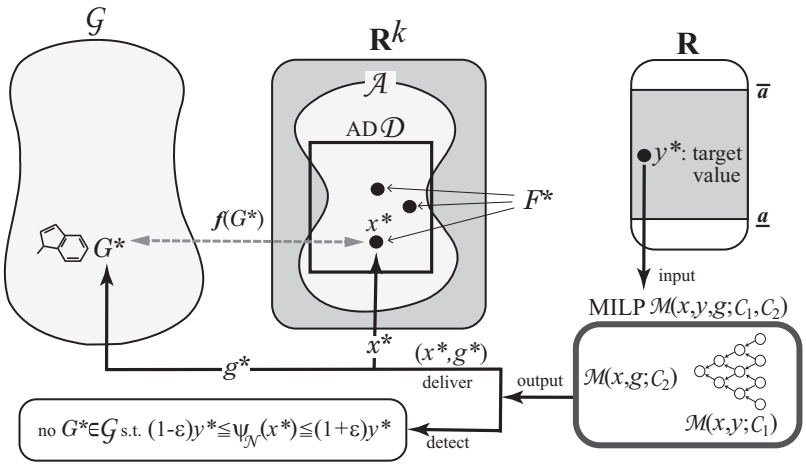


Figure 6. An illustration of Step 4: Given a target value $y^* \in [\underline{a}, \overline{a}]$, solving MILP $\mathcal{M}(x, y, g; \mathcal{C}_1, \mathcal{C}_2)$ either delivers a set F^* of vectors $x^* \in \mathcal{A} \cap \mathcal{D}$ such that $(1 - \varepsilon)y^* \leq \psi_{\mathcal{N}}(x^*) \leq (1 + \varepsilon)y^*$ or detects that no such vector x exists.

Step 5: To solve Problem (II-b), enumerate all graphs $G^* \in \mathcal{G}$ such that $f(G^*) = x^*$ for each vector $x^* \in F^*$. See Figure 7 for an illustration of Step 5.

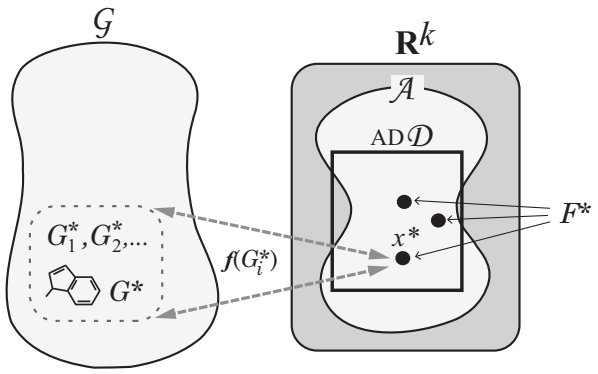


Figure 7. An illustration of Step 5: For each vector $x^* \in F^*$, all chemical graphs $G^* \in \mathcal{G}$ such that $f(G^*) = x^*$ are generated.

In this paper, we set a graph class $\mathcal{G}$ to be the set of rank-2 graphs. In Step 4, we solve an MILP $\mathcal{M}(x, g; \mathcal{C}_2)$ that is formulated on a novel idea of representing rank-2 chemical graphs, as will be discussed in Section 2.3.2. In Step 5, we use branch-and-bound algorithms for enumerating rank-2 chemical compounds [25,26].

2.3. Representing Rank-2 Chemical Graphs

This section introduces a method of modeling rank-2 chemical graphs with different cyclic structures in a unified way and proposes an MILP formulation that represents a rank-2 chemical graph G of n vertices.

2.3.1. Scheme Graphs and Tree-Extensions

Given positive integers n^* and p, a graph with n^* vertices and p edges can be represented as a subgraph of a complete graph K_{n^*} with $n^*(n^*-1)/2$ edges. However, formulating this as an MILP may require to prepare $\Omega((n^*)^2)$ variables and constraints. To reduce the number of variables and constraints in an MILP that represents a rank-2 graph, we decompose a rank-2 graph G into the core and non-core of G so that the core is represented by one of the three rank-2 polymer topologies and the non-core is a collection of trees in which the height is bounded by the core height of G. We do not specify how many subtrees will be attached to each edge in the polymer topology in advance, since otherwise we would need a different MILP for a distinct combination of such assignments of subtrees. Instead we allow each edge in a polymer topology to collect a necessary number of subtrees in our MILP (see the next section for more detail). In this section, we introduce a "scheme graph" to represent three possible rank-2 polymer topologies, an "extension" of the scheme graph to represent the core of a rank-2 graph and a "tree-extension" to represent a combination of the core and non-core of a rank-2 graph, so that any of the three kinds of rank-2 polymer topologies can be selected in a single MILP formulation.

Scheme Graphs

Formally, we define the *scheme graph* for rank 2 to be a pair $(K, \mathcal{E})$ of a multigraph K and an ordered partition $\mathcal{E} = (E_1, E_2, E_3)$ of the edge set $E(K)$. Figure 2d illustrates the scheme graph $(K = (\{u_1, u_2, u_3, u_4\}, E), \mathcal{E} = (E_1, E_2, E_3))$. An edge in E_1 is called a *semi-edge*, an edge in E_2 is called a *virtual edge* and an edge in E_3 is called a *real edge*.

Extensions of Scheme Graphs

Based on the scheme graph $(K, \mathcal{E})$, we construct the core of a rank-2 graph H as an "extension," which is defined as follows (see also Figure 8). The extension H_{core} in Figure 9a An *extension* of the scheme graph $(K, \mathcal{E})$ is defined to be a simple graph obtained from K by using each real edge

$e = uv \in E_3$, by eliminating or replacing each virtual edge $e = uv \in E_2$ (resp., semi-edge $e = uv \in E_1$) with a u, v-path of length at least two (resp., 1) in the core of H, where a u, v-path of length 1 means an edge uv. Figure 9a illustrates an extension H_{core} of the scheme graph $(K, \mathcal{E})$ which is obtained by removing virtual edges $a_4, a_5 \in E_2$ and by replacing semi-edge $a_1 \in E_1$ with a path $(u_{1,1}, v_{1,1}, v_{2,1}, u_{4,1})$, semi-edge $a_2 \in E_1$ with a path $(u_{2,1}, v_{3,1}, v_{4,1}, v_{5,1}, u_{3,1})$ and by using semi-edge $a_3 \in E_1$ and real edges $a_6, a_7 \in E_3$. The extension H_{core} in Figure 9a is isomorphic to the core of the rank-2 graph H in Figure 9b. Observe that each of the least simple graphs $S(M_i)$, $i = 1, 2, 3$ in Figure 2 is obtained as an extension of the scheme graph $(K, \mathcal{E})$ in Figure 2d.

Tree-Extensions

Let $s^* = |V(K)| = 4$ denote the number of vertices in the scheme graph. For non-negative integers a, b and c, we consider a rank-2 graph H such that $\mathrm{cs}(H) = s^* + a = 4 + a$, $\mathrm{ch}(H) = b$ and the maximum degree of a core vertex is at most c. We define an "(a, b, c)-tree-extension" as a minimal supergraph of all such rank-2 graphs H. Formally, the *(a, b, c)-tree-extension* (or a *tree-extension*) is defined to be the graph obtained by augmenting the graph K as follows:

(i) For each vertex $u_s \in V(K)$, $s \in [1, s^*]$, create a copy S_s of the rooted tree $T(c-2, c-1, b)$. For each $s \in [1, s^*]$, let the root of rooted tree S_s be equal to the vertex u_s and denote by $u_{s,i}$ the copy of the i-th vertex of $T(c-2, c-1, b)$ in S_s (see Figure 8a).

(ii) Create a new path $(v_{1,1}, v_{2,1}, \ldots, v_{a,1})$ with a vertices, where the edge between $v_{t,1}$ and $v_{t+1,1}$ is denoted by e_{t+1} (see Figure 8c). For each $t \in [1, a]$, create a copy T_t of the rooted tree $T(c-2, c-1, b)$, let the root of rooted tree T_t be equal to the vertex $v_{1,1}$ and denote by $v_{t,i}$ the copy of the i-th vertex of $T(c-2, c-1, b)$ in T_t (see Figure 8b).

(iii) For every pair (s, t) with $s \in [1, s^*]$ and $t \in [1, a]$, join vertices $u_{s,1}$ and $v_{t,1}$ with an edge $u_{s,1}v_{t,1}$ (see Figure 8c).

Figure 8 illustrates the $(3, 2, 4)$-tree-extension of the scheme graph. We show how a rank-2 graph can be constructed as a subgraph of a tree-extension with some example. Figure 9b illustrates a rank-2 graph H with $n(H) = 21$, $\mathrm{cs}(H) = 9$, $\mathrm{ch}(H) = 2$ and $\theta(H) = 1$, where the maximum degree of a non-core vertex is 3. To prepare a tree-extension so that the graph H can be a subgraph of the tree-extension, we set $\mathrm{cs}^* := \mathrm{cs}(H)$, $a := t^* := \mathrm{cs}^* - s^* = 5$, $b := \mathrm{ch}^* := \mathrm{ch}(H) = 2$ and $c := d_{\max} := 3$. Figure 9c illustrates a subgraph H' of the $(t^* = 5, \mathrm{ch}^* = 2, d_{\max} = 3)$-tree-extension such that H' is isomorphic to the rank-2 graph H.

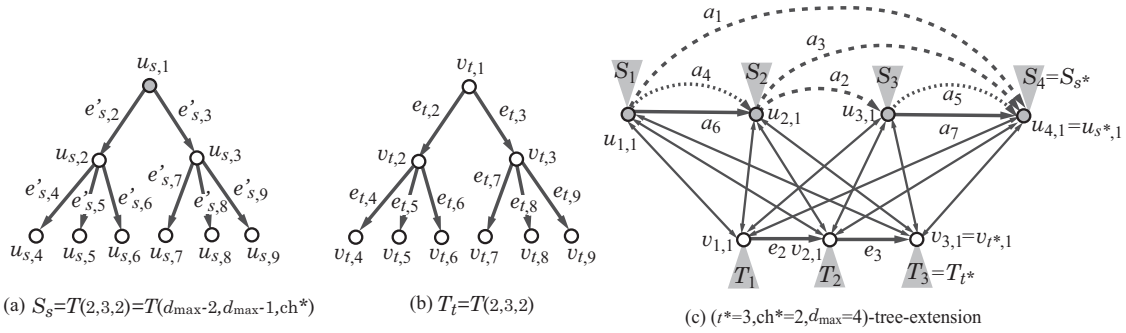


Figure 8. An illustration of a tree-extension, where the vertices in $V(K)$ are depicted with gray circles: (**a**) The structure of the rooted tree S_s rooted at a vertex $u_{s,1}$; (**b**) the structure of the rooted tree T_t rooted at a vertex $v_{t,1}$; (**c**) the (a, b, c)-tree-extension of the scheme graph in Figure 2d for $a = t^* = 3$, $b = \mathrm{ch}^* = 2$ and $c = d_{\max} = 4$.

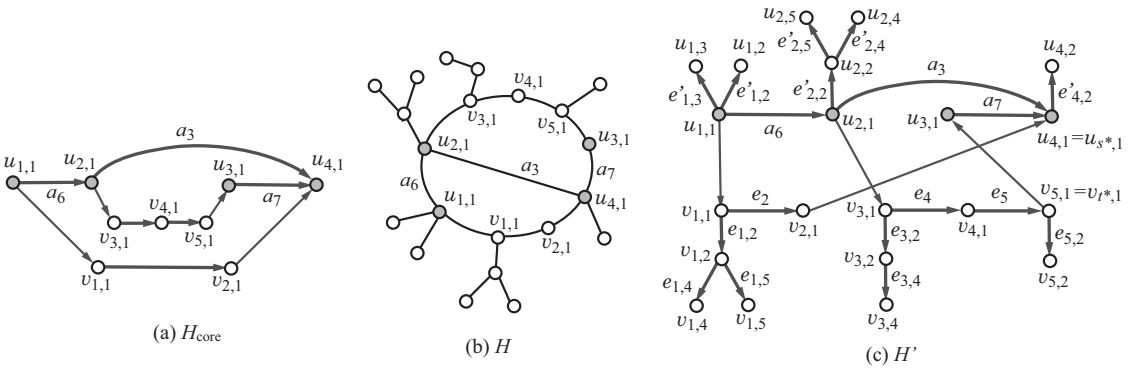


Figure 9. (**a**) An example of an extension of the scheme graph; (**b**) an example of a rank-2 graph H with $n(H) = 21$, $\mathrm{cs}(H) = 9$, $\mathrm{ch}(H) = 2$ and $\theta(H) = 1$, where the labels of some vertices and edges indicate the corresponding vertices and edges in the $(t^*, \mathrm{ch}^*, d_{\max})$-tree-extension for $\mathrm{cs}^* = \mathrm{cs}(H)$, $\mathrm{ch}^* = \mathrm{ch}(H)$, $s^* = 4$, $t^* = \mathrm{cs}^* - s^*$ and $d_{\max} = 3$; (**c**) a subgraph H' of $(t^* = 5, \mathrm{ch}^* = 2, d_{\max} = 3)$-tree-extension isomorphic to the rank-2 graph H in (**b**).

2.3.2. MILPs for Rank-2 Chemical Graphs

We present an outline of our MILP $\mathcal{M}(x, g; \mathcal{C}_2)$ in Step 4 of the framework. For integers $d_{\max}, n^*, \mathrm{cs}^*, \mathrm{ch}^*, \theta^* \in \mathbb{Z}$, let $\mathcal{H}(d_{\max}, n^*, \mathrm{cs}^*, \mathrm{ch}^*, \theta^*)$ denote the set of rank-2 graphs H such that the degree of each core vertex is at most 4, the degree of each non-core vertex is at most $d_{\max}$, $n(H) = n^*$, $\mathrm{cs}(H) = \mathrm{cs}^*$, $\mathrm{ch}(H) = \mathrm{ch}^*$ and $\theta(H) = \theta^*$. In this paper, we obtain the following result.

Theorem 2. *Let Λ be a set of chemical elements, Γ be a set of adjacency-configurations, where $|\Lambda| \leq |\Gamma|$, and $k = 2|\Lambda| + 2|\Gamma| + 15$. Given integers $d_{\max} \in \{3, 4\}$, $n^* \geq 3$, $\mathrm{cs}^* \geq 3$ $\mathrm{ch}^* \geq 0$ and θ^*, there is an MILP $\mathcal{M}(x, g; \mathcal{C}_2)$ that consists of variable vectors $x \in \mathbb{R}^k$ and $g \in \mathbb{R}^q$ for some integer $q = O(|\Gamma| \cdot \mathrm{cs}^* \cdot (d_{\max} - 1)^{\mathrm{ch}^*})$ and a set $\mathcal{C}_2$ of $O(|\Gamma| + \mathrm{cs}^* \cdot (d_{\max} - 1)^{\mathrm{ch}^*})$ constraints on these variables such that: (x^*, g^*) is feasible to $\mathcal{M}(x, g; \mathcal{C}_2)$ if and only if g^* forms a rank-2 chemical graph $G^* = (H, \alpha, \beta) \in \mathcal{G}(\Lambda, \Gamma)$ such that $H \in \mathcal{H}(d_{\max}, n^*, \mathrm{cs}^*, \mathrm{ch}^*, \theta^*)$ and $f(G^*) = x^*$.*

Note that our MILP requires only $O(n^*)$ variables and constraints when the maximum core height of a subtree in the non-core of G^* and $|\Gamma|$ are constant. We formulate an MILP in Theorem 2 so that such a graph H is selected as a subgraph of the scheme graph.

We explain the basic idea of our MILP. Define

$$t^* \triangleq \mathrm{cs}^* - s^*,$$

$$c^* \triangleq |E_1 \cup E_2| \text{ for } (K, \mathcal{E} = (E_1, E_2, E_3)),$$

$$n_{\mathrm{tree}} \triangleq 1 + 2((d_{\max} - 1)^{\mathrm{ch}^*} - 1)/(d_{\max} - 2) \text{ and } n_{\mathrm{in}} \triangleq 1 + 2((d_{\max} - 1)^{\mathrm{ch}^* - 1} - 1)/(d_{\max} - 2),$$

where n_{tree} and n_{in} are the numbers of vertices and non-leaf vertices in the rooted tree $T(d_{\max} - 2, d_{\max} - 1, \mathrm{ch}^*)$, respectively. The MILP mainly consists of the following three types of constraints.

1. Constraints for selecting a rank-2 graph H as a subgraph of the $(t^*, \mathrm{ch}^*, d_{\max})$-tree-extension of the scheme graph $(K, \mathcal{E})$;
2. Constraints for assigning chemical elements to vertices and multiplicity to edges to determine a chemical graph $G = (H, \alpha, \beta)$;
3. Constraints for computing descriptors from the selected rank-2 chemical graph G; and
4. Constraints for reducing the number of rank-2 chemical graphs that are isomorphic to each other but can be represented by the above constraints.

In the constraints of 1, we treat each edge in the tree-extension as a directed edge because describing some condition for H to belong to $\mathcal{H}(d_{\max}, n^*, \mathrm{cs}^*, \mathrm{ch}^*, \theta^*)$ becomes slightly easier than the case of undirected graphs. More formally we prepare the following.

(i) In the scheme graph $(K, \mathcal{E})$, denote the edges in $E_1 \cup E_2 \cup E_3$ by $E_1 = \{a_1, a_2, \ldots, a_{|E_1|}\}$, $E_2 = \{a_{|E_1|+1}, \ldots, a_{c^*}\}$ and $E_3 = \{a_{c^*+1}, \ldots, a_m\}$ (where $c^* = |E_1 \cup E_2|$), and regard each edge $a_i = u_{s,1}u_{s',1} \in E_1 \cup E_2 \cup E_3$ as a directed edge from one end-vertex $u_{s,1}$ to the other end-vertex $u_{s',1}$ with $s < s'$. Let $a(i)$ be a binary variable for each edge a_i, $i \in [1, m]$.

(ii) In each tree S_s (resp., T_t) in the tree-extension, we regard each edge $e'_{s,i}$, $i \geq 2$ in the rooted tree S_s, $s \in [1, s^*]$ (resp., $e_{t,i}$, $i \geq 2$ in the rooted tree T_t, $t \in [1, t^*]$) as a directed edge from vertex $u_{s,\mathrm{prt}(i)}$ to vertex $u_{s,i}$ (resp., from vertex $v_{t,\mathrm{prt}(i)}$ to vertex $v_{t,i}$). Let $u(s,i)$ (resp., $v(t,i)$) be a binary variable for vertex $u_{s,i}$, $s \in [1, s^*]$ (resp., $t \in [1, t^*]$) and $i \in [1, n_{\mathrm{tree}}]$;

(iii) In the path P_{t^*} consisting of the roots of trees T_t, $[t \in 1, t^*]$, we regard each edge e_t, $t \in [2, t^*]$ as a directed edge from vertex $v_{t-1,1}$ to vertex $v_{t,1}$; and

(iv) We regard each edge $u_{s,1}v_{t,1}$ for $s \in [1, s^*]$ and $t \in [1, t^*]$ as two directed edges, one directed from vertex $u_{s,1}$ to vertex $v_{t,1}$ and the other directed oppositely. Let $e(s,t)$ (resp., $e(t,s)$) be a binary variable of directed edge $(u_{s,1}, v_{t,1})$ (resp., $(v_{t,1}, u_{s,1})$).

Based on these, we include constraints with some more additional variables so that a selected subgraph H is a connected rank-2 graph. See constraints Equations (A10) to (A42) in Appendix A for the details.

In the constraints of 2, we prepare an integer variable $\widetilde{\alpha}(u)$ for each vertex u in the tree-extension that represents the chemical element $\alpha(u) \in \Lambda$ if u is in a selected graph H (or $\widetilde{\alpha}(u) = 0$ otherwise) and an integer variable $\widetilde{\beta}(e) \in [0, 3]$ (resp., $\widehat{\beta}(e) \in [0, 3]$) for each edge e (resp., $e = e(s,t)$ or $e(t,s)$, $s \in [1, s^*]$, $t \in [1, t^*]$) in the tree-extension that represents the multiplicity $\beta(e) \in [1, 3]$ if e is in a selected graph H (or $\widetilde{\beta}(e)$ or $\widehat{\beta}(e)$ takes 0 otherwise). This determines a chemical graph $G = (H, \alpha, \beta)$. Also we include constraints for a selected chemical graph G to satisfy the valence condition $(\alpha(u), \alpha(v), \beta(uv)) \in \Gamma$ for each edge $uv \in E$. See constraints Equations (A43) to (A61) in Appendix A for the details.

In the constraints of 3, we introduce a variable for each descriptor and constraints with some more variables to compute the value of each descriptor in $f(G)$ for a selected chemical graph G. See constraints Equations (A62) to (A113) in Appendix A for the details.

With constraints 1 to 3, our MILP formulation already represents a rank-2 chemical graph G and a feature vector $x \in \mathbb{R}^k$ so that $x = f(G)$ holds. In the constraints of 4, we include some additional constraints so that the search space required for an MILP solver to solve an instance of our MILP problem is reduced. For this, we consider a graph-isomorphism of rooted subtrees of each tree S_s or T_s and define a canonical form among subtrees that are isomorphic to each other. We try to eliminate a chemical graph G that has a subtree in S_s or T_s that is not a canonical form. See constraints Equations (A114) to (A119) in Appendix A for the details.

3. Results

We implemented our method of Steps 1 to 5 for inferring rank-2 chemical graphs and conducted experiments to evaluate the computational efficiency for three chemical properties π: octanol/water partition coefficient (Kow), melting point (MP), and boiling point (BP). We executed the experiments on a PC with Intel Core i5 1.6 GHz CPU and 8GB of RAM running under the Mac OS operating system version 10.14.6. We show 2D drawings of some of the inferred chemical graphs, where ChemDoodle version 10.2.0 is used for constructing the drawings.

Results on Phase 1.

Step 1. We set a graph class $\mathcal{G}$ to be the set of all rank-2 chemical graphs. For each property $\pi \in \{\mathrm{Kow}, \mathrm{MP}, \mathrm{BP}\}$, we select a set Λ of chemical elements and collected a data set D_π on rank-2 chemical graphs over Λ provided by HSDB from PubChem. To construct the data set, we eliminated chemical

compounds that have at most three carbon atoms or contain a charged element such as $\mathtt{N}^+$ or an element $\mathtt{a} \in \Lambda$ in which the valence is different from our setting of valence function val.

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

- π: one of the chemical properties Kow, MP and BP;
- $|D_\pi|$: the size of data set D_π for property π;
- Λ: the set of chemical elements over data set D_π (hydrogen atoms are added at the final stage);
- $|\Gamma|$: the number of tuples in Γ;
- $[\underline{n}, \overline{n}]$: the minimum and maximum number $n(G)$ of non-hydrogen atoms over data set D_π;
- $[\underline{\mathrm{cs}}, \overline{\mathrm{cs}}]$, $[\underline{\mathrm{ch}}, \overline{\mathrm{ch}}]$: the minimum and maximum core size and core height over chemical compounds in D_π, respectively;
- $[\underline{\theta}, \overline{\theta}]$: the minimum and maximum values of the topological parameter $\theta(G)$ over data set D_π; and
- $[\underline{a}, \overline{a}]$: the minimum and maximum values of $a(G)$ in π over data set D_π.

Table 1. Results of Step 1 in Phase 1.

π	$\lvert D_\pi \rvert$	Λ	$\lvert\Gamma\rvert$	$[\underline{n}, \overline{n}]$	$[\underline{\mathrm{cs}}, \overline{\mathrm{cs}}]$	$[\underline{\mathrm{ch}}, \overline{\mathrm{ch}}]$	$[\underline{\theta}, \overline{\theta}]$	$[\underline{a}, \overline{a}]$
Kow	93	C,N,O	9	[9, 31]	[7, 16]	[0, 13]	[−5, 3]	[−3.7, 12.2]
MP	63	C,N,O	7	[9, 31]	[7, 17]	[0, 4]	[−6, 3]	[−80, 300]
BP	45	C,N,O,S,P,Cl	9	[9, 25]	[7, 15]	[0, 7]	[−4, 3]	[155, 420]

Step 2. We used a feature function f that consists of the descriptors defined in Section 2.1.

Step 3. We used `scikit-learn` version 0.21.6 with Python 3.7.4 to construct ANNs $\mathcal{N}$ where the tool and activation function are set to be MLPRegressor and ReLU, respectively. We tested several different architectures of ANNs for each chemical property. To evaluate the performance of the resulting prediction function $\psi_{\mathcal{N}}$ with cross-validation, we partition a given data set D_π into five subsets $D_\pi^{(i)}$, $i \in [1,5]$ randomly, where $D_\pi \setminus D_\pi^{(i)}$ is used for a training set and $D_\pi^{(i)}$ is used for a test set in five trials $i \in [1,5]$. For a set $\{y_1, y_2, \ldots, y_N\}$ of observed values and a set $\{\psi_1, \psi_2, \ldots, \psi_N\}$ of predicted values, we define the coefficient of determination to be $\mathrm{R}^2 \triangleq 1 - \frac{\sum_{j\in[1,N]}(y_j-\psi_j)^2}{\sum_{j\in[1,N]}(y_j-\overline{y})^2}$, where $\overline{y} = \frac{1}{N}\sum_{j\in[1,N]} y_j$. Table 2 shows the results on Steps 2 and 3, where

- k: the number of descriptors for the chemical compounds in data set D_π for property π;
- Activation: the choice of activation function;
- Architecture: $(a, b, 1)$ consists of an input layer with a nodes, a hidden layer with b nodes, and an output layer with a single node, where a is equal to the number of descriptors;
- L-time: the average time (sec.) to construct ANNs for each trial;
- test R^2 (ave.): the average of coefficient of determination over the five test sets; and
- test R^2 (best): the largest value of coefficient of determination over the five test sets.

For each chemical property π, we selected the ANN $\mathcal{N}$ that attained the best test R^2 score among the five ANNs to formulate an MILP $\mathcal{M}(x, y, z; \mathcal{C}_1)$ in the second phase.

Table 2. Results of Steps 2 and 3 in Phase 1.

π	k	Activation	Architecture	L-Time	Test R^2 (ave.)	(Best)
Kow	37	relu	(37,10,1)	3.92	0.866	0.964
MP	33	relu	(33,10,1)	21.68	0.805	0.916
BP	43	relu	(43,10,1)	11.88	0.802	0.947

Results on Phase 2.

We implemented Steps 4 and 5 in Phase 2 as follows.

Step 4. In this step, we solve the MILP $\mathcal{M}(x, y, g; \mathcal{C}_1, \mathcal{C}_2)$ formulated based on the ANN $\mathcal{N}$ obtained in Phase 1. To solve an MILP in Step 4, we use CPLEX version 12.10. In our experiment, we choose a target value $y^* \in [\underline{a}, \overline{a}]$ and fix or bound some descriptors in our feature vector as follows:

- Fix variable θ that represents the polymer parameter $\theta(H)$ to be each integer in $\{-2, 0, 2\}$;
- Set $d_{\max}$ to be each of 3 and 4;
- Fix n^* to be some four integers in $\{15, 19, 20, 25, 30\}$ for $\theta \in \{-2, 0\}$ and $\{15, 19, 20, 22, 25\}$ for $\theta = 2$;
- Choose three integers from $[7, 16]$ and fix cs* to be each of the three integers;
- Fix ch* to be each of the four integers in $[2, 5]$.

Based on the above setting, we generated 12 instances for each n^*. We set $\varepsilon = 0.02$ in Step 4.

Tables 3–8 show the results of Step 4 for $d_{\max} = 3$ and 4, respectively, where we denote the following:

- y^*_π: a target value in $[\underline{a}, \overline{a}]$ for a property π;
- n^*: a specified number of vertices in $[\underline{n}, \overline{n}]$;
- $|F^*|/$#I: #I means the number of MILP instances in Step 4 (where #I = 12), and $|F^*|$ means the size of set F^* of vectors x^* generated from all feasible instances among the #I MILP instances in Step 4;
- IP-time: the average time (sec.) to solve one of the #I MILP instances to find a set F^* of vectors x^*.

Figure 10a–c illustrate some rank-2 chemical graphs G^* with $\theta(G^*) = -2$ constructed from the vector g^* obtained by solving the MILP in Step 4.

Figure 11a–c illustrate some rank-2 chemical graphs G^* with $\theta(G^*) = 0$ constructed from the vector g^* obtained by solving the MILP in Step 4.

Table 3. Results of Steps 4 and 5 with $d_{\max} = 3$ and $\theta = -2$.

π	y^*	n^*	$\lvert F^*\rvert$/#I	IP-Time	#G^*	G-Time
Kow	5	15	12/12	9.96	100	2236.0
Kow	5	20	12/12	30.38	12	>1 h
Kow	5	25	12/12	47.57	12	>1 h
Kow	5	30	12/12	69.38	12	>1 h
MP	150	15	12/12	9.52	100	2069.0
MP	150	20	12/12	22.79	12	>1 h
MP	150	25	12/12	47.20	12	>1 h
MP	150	30	12/12	66.90	12	>1 h
BP	250	15	11/12	9.50	100	103.5
BP	250	19	12/12	19.08	12	>1 h
BP	250	22	12/12	25.78	12	>1 h
BP	250	25	12/12	67.64	12	>1 h

Table 4. Results of Steps 4 and 5 with $d_{\max} = 4$ and $\theta = -2$.

π	y^*	n^*	$\lvert F^*\rvert$/#I	IP-Time	#G^*	G-Time
Kow	5	15	11/12	31.84	100	413.8
Kow	5	20	12/12	69.65	12	>1 h
Kow	5	25	12/12	144.20	11	>1 h
Kow	5	30	12/12	352.01	12	>1 h
MP	150	15	9/12	20.68	100	947.4
MP	150	20	11/12	73.73	11	>1 h
MP	150	25	9/12	140.09	9	>1 h
MP	150	30	12/12	304.04	12	>1 h
BP	250	15	7/12	28.51	100	232.7
BP	250	19	11/12	82.01	11	>1 h
BP	250	22	12/12	150.55	12	>1 h
BP	250	25	12/12	239.84	12	>1 h

Table 5. Results of Steps 4 and 5 with $d_{\max} = 3$ and $\theta = 0$.

π	y^*	n^*	$\lvert F^* \rvert$/#I	IP-Time	#G^*	G-Time
Kow	5	15	12/12	11.00	100	121.1
Kow	5	20	12/12	25.64	12	>1 h
Kow	5	25	12/12	38.79	12	>1 h
Kow	5	30	12/12	49.65	12	>1 h
MP	150	15	12/12	8.45	100	373.4
MP	150	20	12/12	18.94	12	>1 h
MP	150	25	12/12	37.13	12	>1 h
MP	150	30	12/12	44.745	4	>1 h
BP	250	15	9/12	8.450	100	74.2
BP	250	19	11/12	16.31	11	>1 h
BP	250	22	12/12	21.71	12	>1 h
BP	250	25	12/12	45.80	12	>1 h

Table 6. Results of Steps 4 and 5 with $d_{\max} = 4$ and $\theta = 0$.

π	y^*	n^*	$\lvert F^* \rvert$/#I	IP-Time	#G^*	G-Time
Kow	5	15	9/12	36.33	100	23.2
Kow	5	20	12/12	82.01	12	>1 h
Kow	5	25	12/12	138.96	12	>1 h
Kow	5	30	12/12	292.79	12	>1 h
MP	150	15	9/12	19.89	100	557.6
MP	150	20	11/12	63.62	11	>1 h
MP	150	25	12/12	112.49	12	>1 h
MP	150	30	12/12	171.11	12	>1 h
BP	250	15	3/12	34.60	100	11.2
BP	250	19	6/12	203.65	6	>1 h
BP	250	22	9/12	218.07	9	>1 h
BP	250	25	11/12	783.80	11	>1 h

Table 7. Results of Steps 4 and 5 with $d_{\max} = 3$ and $\theta = 2$.

π	y^*	n^*	$\lvert F^* \rvert$/#I	IP-Time	#G^*	G-Time
Kow	5	15	12/12	11.64	100	1386.7
Kow	5	20	12/12	23.84	12	>1 h
Kow	5	25	12/12	33.71	12	>1 h
Kow	5	30	12/12	61.85	12	>1 h
MP	150	15	12/12	9.80	100	1614.3
MP	150	20	12/12	20.15	12	>1 h
MP	150	25	12/12	36.42	12	>1 h
MP	150	30	12/12	40.58	12	>1 h
BP	250	15	11/12	10.25	100	1756.1
BP	250	19	12/12	16.02	12	>1 h
BP	250	22	12/12	23.63	12	>1 h
BP	250	25	12/12	63.84	12	>1 h

Table 8. Results of Steps 4 and 5 with $d_{\max} = 4$ and $\theta = 2$.

π	y^*	n^*	$\lvert F^* \rvert$/#I	IP-Time	#G^*	G-Time
Kow	5	15	11/12	28.15	100	20.3
Kow	5	20	12/12	71.90	12	>1 h
Kow	5	25	12/12	112.71	12	>1 h
Kow	5	30	12/12	267.21	12	>1 h
MP	150	15	9/12	22.53	100	2748.1
MP	150	20	11/12	53.44	11	>1 h
MP	150	25	12/12	143.33	12	>1 h
MP	150	30	12/12	220.63	12	>1 h
BP	250	15	6/12	27.33	100	254.2
BP	250	19	9/12	75.50	9	>1 h
BP	250	22	11/12	133.01	11	>1 h
BP	250	25	12/12	228.75	12	>1 h

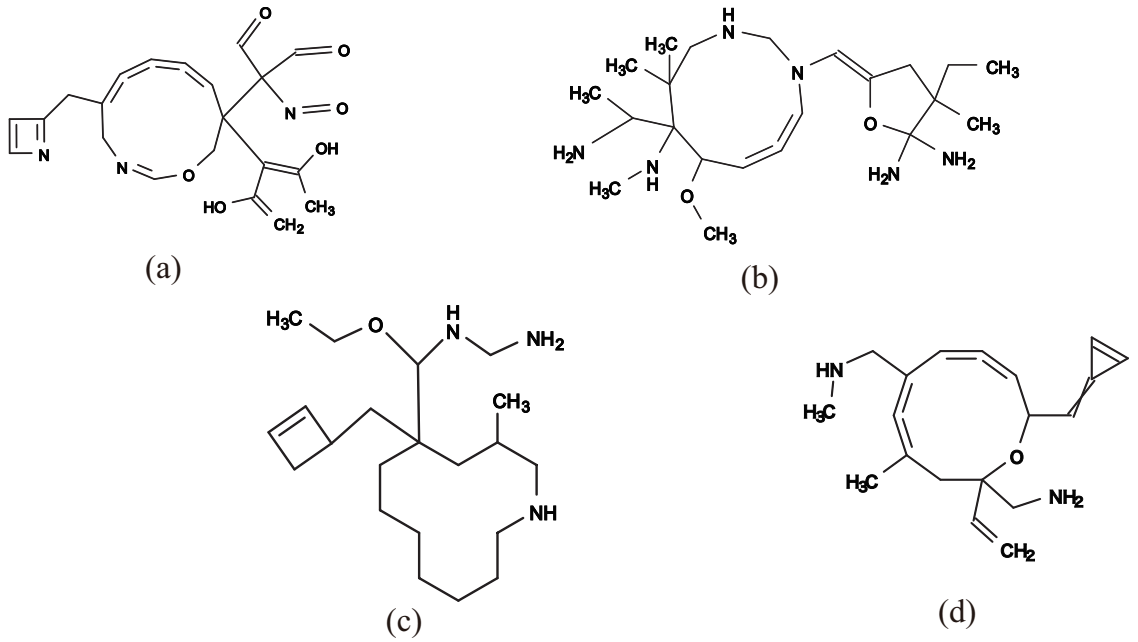


Figure 10. An illustration of inferred rank-2 chemical graphs G^* with $\theta = -2$: (**a**) $y^*_{\mathrm{Kow}} = 5$, $\theta = -2$, $n = 30$, core size = 16, core height = 3, $d_{\max} = 4$; (**b**) $y^*_{\mathrm{Mp}} = 250$, $\theta = -2$, $n = 30$, core size = 16, core height= 2, $d_{\max} = 3$; (**c**) $y^*_{\mathrm{Bp}} = 150$, $\theta = -2$, $n = 25$, core size = 17, core height = 4, $d_{\max} = 3$; (**d**) $y^*_{\mathrm{Kow}} = 5$, $y^*_{\mathrm{Mp}} = 150$, $y^*_{\mathrm{Bp}} = 250$, $\theta = -2$, $n = 22$, core size = 14, core height = 3, $d_{\max} = 3$.

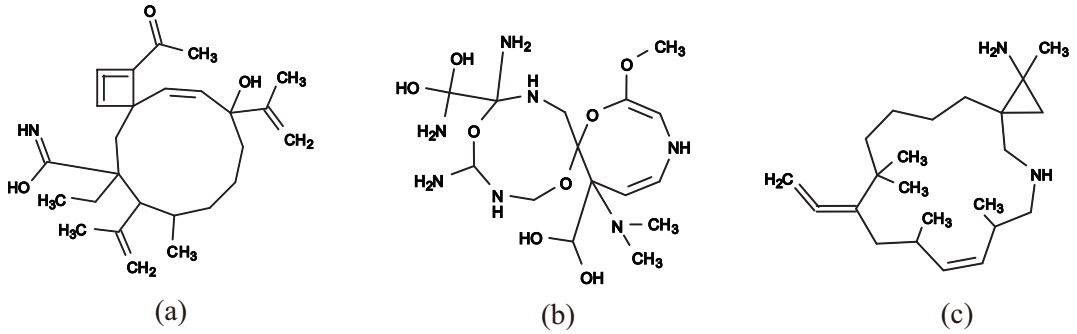


Figure 11. An illustration of inferred rank-2 chemical graphs G^*: (**a**) $y^*_{\mathrm{Kow}} = 5$, $\theta = 0$, $n = 30$, core size = 14, core height = 2, $d_{\max} = 3$; (**b**) $y^*_{\mathrm{Mp}} = 250$, $\theta = 0$, $n = 30$, core size = 16, core height = 2, $d_{\max} = 4$; (**c**) $y^*_{\mathrm{Bp}} = 150$, $\theta = 0$, $n = 25$, core size = 17, core height = 2, $d_{\max} = 3$.

Figure 12a–c illustrate some rank-2 chemical graphs G^* with $\theta(G^*) = 2$ constructed from the vector g^* obtained by solving the MILP in Step 4.

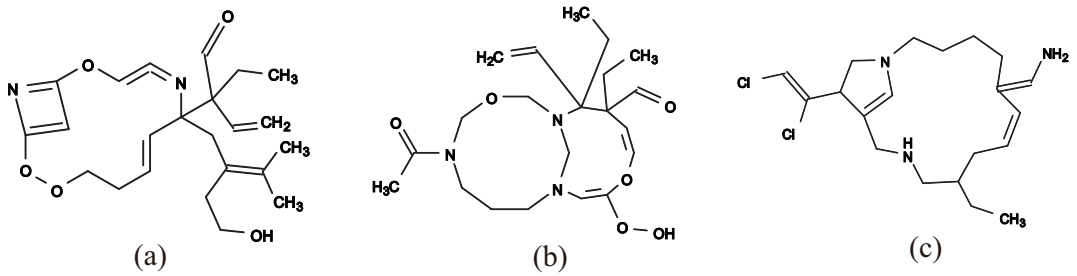


Figure 12. An illustration of inferred rank-2 chemical graphs G^*: (**a**) $y^*_{\mathrm{Kow}} = 5$, $\theta = 2$, $n = 30$, core size = 15, core height = 5, $d_{\max} = 4$; (**b**) $y^*_{\mathrm{Mp}} = 250$, $\theta = 2$, $n = 30$, core size = 17, core height = 2, $d_{\max} = 3$; (**c**) $y^*_{\mathrm{Bp}} = 150$, $\theta = 2$, $n = 25$, core size = 17, core height = 3, $d_{\max} = 3$.

Step 5. In this step, we modified the algorithms proposed by Tamura et al. [25] and Yamashita et al. [26] to enumerate all rank-2 graphs $G^* \in \mathcal{G}$ such that $f(G^*) = x^*$ for each $x^* \in F^*$. We stop the execution when either the total number of graphs inferred over all vectors $x^* \in F^*$ exceeds 100 or the execution time exceeds one hour.

Tables 3–8 show the results on Step 5 for $d_{\max} = 3$ and 4, respectively,

- #G^*: the number of all (or up to 100) rank-2 chemical graphs G^* that are computed under 1 h time limit in Step 5, where $f(G^*) = x^*$ for some $x^* \in F^*$. (Note that $|F^*|$ such graphs G^* have been found in Step 4, and Figures 10–12 illustrate some of such graphs G^*.);
- G-time: the running time (sec.) to execute Step 5, where ">1 h" means that the execution time exceeds the limit.

We also conducted some additional experiments to demonstrate that our MILP-based method is flexible to control conditions on the inference of chemical graphs. In Step 3, we constructed an ANN $\mathcal{N}_\pi$ for each of the three chemical properties $\pi \in \{\text{Kow}, \text{MP}, \text{BP}\}$, and formulated the inverse problem of each ANN $\mathcal{N}_\pi$ as an MILP $\mathcal{M}_\pi$. Since the set of descriptors is common to all three properties Kow, MP, and BP, it is possible to infer a rank-2 chemical graph G^* that satisfies a target value y^*_π for each of the three properties at the same time (if one exists). We specify the size of graph so that n := 22, core size := 14, core height := 3, $\theta := -2$ and $d_{\max} := 3$, and set target values with $y^*_{\text{Kow}} := 5$, $y^*_{\text{Mp}} := 150$ and $y^*_{\text{Bp}} := 250$ in an MILP that consists of the three MILPs $\mathcal{M}_{\text{Kow}}$, $\mathcal{M}_{\text{Mp}}$ and $\mathcal{M}_{\text{Bp}}$. The MILP was solved in 268.11 (sec) and we obtained a rank-2 chemical graph G^* illustrated in Figure 10d.

4. Discussion

In this paper, we proposed a new method for the inverse QSAR/QSPR to rank-2 chemical graphs by significantly enhancing the framework due to Azam et al. [18], Zhang et al. [20], and Ito et al. [21], and implemented it for inferring rank-2 chemical graphs using the algorithms for enumerating rank-2 chemical graphs due to Tamura et al. [25] and Yamashita et al. [26]. From the results on some computational experiments, we observe that the proposed method runs efficiently for an instance with $n^* \leq 30$ non-hydrogen atoms up to Step 4 and an instance with $n^* \leq 15$ non-hydrogen atoms up to Step 5. Due to this development, the ratio of chemical compounds covered in the PubChem database increased from 16.26% to 44.5%. It is left as future work to apply our new method for the inverse QSAR/QSPR to a wider class of graphs. The ratio of the number of chemical graphs with rank at most 3 (resp., 4) to the number of all chemical graphs in database PubChem is 68.8% (resp., 84.7%). Among rank-4 chemical compounds, Remdesivir $\mathtt{C_{27}H_{35}N_6O_8P}$, an antiviral medication, which is being studied as a possible post-infection treatment for COVID-19, has a chemical graph G with $\mathrm{r}(G) = 4$, $n(G) = 42$, $\mathrm{cs}(G) = 24$, and $\mathrm{ch}(G) = 8$. The number of polymer topologies with rank 3 (resp., 4) such that the maximum degree is at most 4 is 12 (resp., 73). Our MILP formulation can be easily extended to the case of rank 3 or 4 by replacing the current set of constraints for the scheme graph with a set of those for a new scheme graph that is designed for rank-3 or -4 polymer topologies.

Author Contributions: Conceptualization, H.N. and T.A.; methodology, H.N.; software, J.Z., C.W., and A.S.; validation, J.Z., C.W., A.S., and H.N.; formal analysis, H.N.; data resources, H.N. and T.A.; writing—original draft preparation, H.N.; writing—review and editing, T.A.; project administration, H.N.; funding acquisition, T.A. All authors have read and agreed to the published version of the manuscript.

Funding: H.N. and T.A. were partially supported by the Japan Society for the Promotion of Science, Japan, under Grant #18H04113.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. All Constraints in an MILP Formulation for Rank-2 Chemical Graphs

To formulate an MILP that represents a chemical graph $G = (H, \alpha, \beta)$, we distinguish a tuple $(\mathtt{a}, \mathtt{b}, k)$ from a tuple $(\mathtt{b}, \mathtt{a}, k)$. For a tuple $\gamma = (\mathtt{a}, \mathtt{b}, k) \in \Lambda \times \Lambda \times \{1, 2, 3\}$, let $\overline{\gamma}$ denote the tuple $(\mathtt{b}, \mathtt{a}, k)$. Let $\Gamma_< \triangleq \{\overline{\gamma} \mid \gamma \in \Gamma_>\}$. We call a tuple $\gamma = (\mathtt{a}, \mathtt{b}, k) \in \Lambda \times \Lambda \times \{1, 2, 3\}$ *proper* if

$$k \leq \min\{\mathrm{val}(\mathtt{a}), \mathrm{val}(\mathtt{b})\} \text{ and } k \leq \max\{\mathrm{val}(\mathtt{a}), \mathrm{val}(\mathtt{b})\} - 1,$$

where the latter is assumed because otherwise G must consist of two atoms of $\mathtt{a} = \mathtt{b}$. Assume that each tuple $\gamma \in \Gamma$ is proper. Let ϵ be a fictitious chemical element that represents null, call a tuple $(\mathtt{a}, \mathtt{b}, 0)$ with $\mathtt{a}, \mathtt{b} \in \Lambda \cup \{\epsilon\}$ *fictitious*, and define Γ_0 to be the set of all fictitious tuples; i.e., $\Gamma_0 = \{(\mathtt{a}, \mathtt{b}, 0) \mid \mathtt{a}, \mathtt{b} \in \Lambda \cup \{\epsilon\}\}$. To represent chemical elements $\mathtt{e} \in \Lambda \cup \{\epsilon\} \cup \Gamma$ in an MILP, we encode these elements $\mathtt{e}$ into some integers denoted by $[\mathtt{e}]$. Assume that, for each element $\mathtt{a} \in \Lambda$, $[\mathtt{a}]$ is a positive integer and that $[\epsilon] = 0$.

Appendix A.1. Applicability Domain

We use the range-based method to define an applicability domain for our method. For this, we find the range (the minimum and maximum) of each descriptor over all relevant chemical compounds and represent each range as a set of linear constraints in the constraint set $\mathcal{C}_1$ of our MILP formulation. Recall that D_π stands for a set of chemical graphs used for constructing a prediction function. However, the number of examples in D_π may not be large enough to capture a general feature on the structure of chemical graphs. For this, we also use some data set from the whole set DB of chemical graphs in a database. Let $\mathrm{DB}_{\mathcal{G}}^{(i)}$ denote the set of chemical graphs $G \in \mathrm{DB} \cap \mathcal{G}$ such that $n(G) = i$ for each integer $i \geq 1$. Formally the set of variables and constraints is given as follows.

AD constraints in $\mathcal{C}_1$:
constants:
Integers $\mathrm{cs}^* \geq 3$ and $\mathrm{ch}^* \geq 1$; An integer $d_{\max} \in \{3, 4\}$;
An integer $n^* \in [\mathrm{cs}^* + 1, \mathrm{cs}^* \cdot (d_{\max} - 1)^{\mathrm{ch}^*}]$;
variables for descriptors in x:
A real variable $\kappa_1 \geq 0$: κ_1 represents $\kappa_1(H)$;
$\mathrm{dg}(i) \in [0, n^*]$ $(i \in [1, 4])$: $\mathrm{dg}(i)$ represents the number of vertices of degree i in H;
$\mathrm{Mass} \in \mathbb{Z}$: Mass represents $\sum_{v \in V} \mathrm{mass}^*(\alpha(v))$;
$\mathrm{ce}^{\mathrm{co}}(\mathtt{a}) \in [0, n^*]$, $\mathtt{a} \in \Lambda$: $\mathrm{ce}^{\mathrm{co}}(\mathtt{a})$ represents the number of vertices of chemical element $\mathtt{a}$ in the core of H;
$\mathrm{ce}^{\mathrm{nc}}(\mathtt{a}) \in [0, n^*]$, $\mathtt{a} \in \Lambda$: $\mathrm{ce}^{\mathrm{nc}}(\mathtt{a})$ represents the number of vertices of chemical element $\mathtt{a}$ in the non-core of H;
$\mathrm{b}^{\mathrm{co}}(k) \in [0, 2n^*]$, $k \in [1, 3]$: $\mathrm{b}^{\mathrm{co}}(k)$ represents the number of k-bonds in the core of H;
$\mathrm{b}^{\mathrm{nc}}(k) \in [0, 2n^*]$, $k \in [1, 3]$: $\mathrm{b}^{\mathrm{nc}}(k)$ represents the number of k-bonds in the non-core of H;
$\mathrm{ac}^{\mathrm{co}}(\gamma) \in [0, n^*]$, $\gamma \in \Gamma_< \cup \Gamma_=$: $\mathrm{ac}^{\mathrm{co}}(\gamma)$ represents the number of core edges in H that are assigned tuple $\gamma \in \Gamma_<$;
$\mathrm{ac}^{\mathrm{nc}}(\gamma) \in [0, n^*]$, $\gamma \in \Gamma_< \cup \Gamma_=$: $\mathrm{ac}^{\mathrm{nc}}(\gamma)$ represents the number of non-core edges in H that are assigned tuple $\gamma \in \Gamma_<$;

constraints:

$$n^* \min_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\kappa_1(G)}{n(G)} \le \kappa_1 \le n^* \max_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\kappa_1(G)}{n(G)}, \tag{A1}$$

$$n^* \min_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\mathrm{dg}_i(G)}{n(G)} \le \mathrm{dg}(i) \le n^* \max_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\mathrm{dg}_i(G)}{n(G)}, \quad i\in[1,4], \tag{A2}$$

$$n^* \min_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \overline{\mathrm{ms}}(G) \le \mathrm{Mass} \le n^* \max_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \overline{\mathrm{ms}}(G), \tag{A3}$$

$$n^* \min_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\mathrm{ce}_{\mathtt{a}}^{\mathrm{co}}(G)}{n(G)} \le \mathrm{ce}^{\mathrm{co}}(\mathtt{a}) \le n^* \max_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\mathrm{ce}_{\mathtt{a}}^{\mathrm{co}}(G)}{n(G)}, \quad \mathtt{a}\in\Lambda, \tag{A4}$$

$$n^* \min_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\mathrm{ce}_{\mathtt{a}}^{\mathrm{nc}}(G)}{n(G)} \le \mathrm{ce}^{\mathrm{nc}}(\mathtt{a}) \le n^* \max_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\mathrm{ce}_{\mathtt{a}}^{\mathrm{nc}}(G)}{n(G)}, \quad \mathtt{a}\in\Lambda, \tag{A5}$$

$$(n^*+1) \min_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{b_k^{\mathrm{co}}(G)}{n(G)+1} \le \mathrm{b}^{\mathrm{co}}(k) \le (n^*+1) \max_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{b_k^{\mathrm{co}}(G)}{n(G)+1}, \quad k\in[2,3], \tag{A6}$$

$$(n^*+1) \min_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{b_k^{\mathrm{nc}}(G)}{n(G)+1} \le \mathrm{b}^{\mathrm{nc}}(k) \le (n^*+1) \max_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{b_k^{\mathrm{nc}}(G)}{n(G)+1}, \quad k\in[2,3], \tag{A7}$$

$$(n^*+1) \min_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\mathrm{ac}_\gamma^{\mathrm{co}}(G)}{n(G)+1} \le \mathrm{ac}^{\mathrm{co}}(\gamma) \le (n^*+1) \max_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\mathrm{ac}_\gamma^{\mathrm{co}}(G)}{n(G)+1}, \quad \gamma\in\Gamma, \tag{A8}$$

$$(n^*+1) \min_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\mathrm{ac}_\gamma^{\mathrm{nc}}(G)}{n(G)+1} \le \mathrm{ac}^{\mathrm{nc}}(\gamma) \le (n^*+1) \max_{G\in D_\pi\cup \mathrm{DB}_{\mathcal{G}}^{(n^*)}} \frac{\mathrm{ac}_\gamma^{\mathrm{nc}}(G)}{n(G)+1}, \quad \gamma\in\Gamma. \tag{A9}$$

In the following, we derive an MILP $\mathcal{M}(x,g;\mathcal{C}_2)$ that satisfies the condition in Theorem 2. Let $d_{\max}\in\{3,4\}$, $n^*\ge 3$, $\mathrm{cs}^*\ge 3$ $\mathrm{ch}^*\ge 0$ and θ^* be given integers. We describe the set $\mathcal{C}_2$ with several sets of constraints.

Appendix A.2. Construction of Scheme Graph and Tree-Extension

We infer a subgraph H such that the maximum degree is $d_{\max}\in\{3,4\}$, $n(H)=n^*$, $\mathrm{cs}(H)=\mathrm{cs}^*$ and $\mathrm{ch}(H)=\mathrm{ch}^*$. For this, we first construct the $(t^*,\mathrm{ch}^*,d_{\max})$-tree-extension of the scheme graph $(K=(V_K=\{u_1,\dots,u_{s^*}\},E_K=\{a_1,a_2,\dots,a_m\}),\mathcal{E}=(E_1,E_2,E_3))$. We use the following notations: For $j\in[1,3]$ and $s\in[1,s^*]$, let $E_j^+(s)$ (resp., $E_j^-(s)$) denote the set of indices i of edges $a_i\in E_i$ such that the tail (resp., head) of a_i is $u_{s,1}$. Let $E_{j,k}^+(s)\triangleq E_j^+(s)\cup E_k^+(s)$, $E_{j,k}^-(s)\triangleq E_j^-(s)\cup E_k^-(s)$, $E_j(s)\triangleq E_j^+(s)\cup E_j^-(s)$ and $E_{j,k}(s)\triangleq E_j(s)\cup E_k(s)$.

As described in Section 2.3.1, some edge $a(i)\in E_1\cup E_2$ may be replaced with a subpath P_i of $(v_{1,1},v_{1,2},\dots,v_{t^*,1})$, which consists of the roots of trees $T_1,T_2,\dots,T_{t^*}$. We assign color i to the vertices in such a subpath P_i by setting a variable $\chi(t)$ of each vertex $v_{t,1}\in V(P_i)$ to be i. For each edge $u_{s,1}v_{t,1}$, we prepare a binary variable $e(s,t)$ to denote that edge $u_{s,1}v_{t,1}$ is used (resp., not used) in a selected graph H when $e(s,t)=1$ (resp., $e(s,t)=0$). We also include constraints necessary for the variables to satisfy a degree condition at each of the vertices $u_{s,1}$, $s\in[1,s^*]$ and $v_{t,1}$, $t\in[1,t^*]$.

constants:
Integers $s^*=|V_K|$, $c^*=|E_1\cup E_2|$, cs^* $(\ge s^*)$, n^* $(\ge \mathrm{cs}^*)$ and $\mathrm{ch}^*\ge 0$;
$\underline{d}^+(s)$, $s\in[1,s^*]$: a lower bound on the out-degree of vertex $u_{s,1}$ in H;
$\underline{d}^-(s)$, $s\in[1,s^*]$: a lower bound on the in-degree of vertex $u_{s,1}$ in H;
$\overline{d}^+(s)$, $s\in[1,s^*]$: an upper bound on the out-degree of vertex $u_{s,1}$ in H;
$\overline{d}^-(s)$, $s\in[1,s^*]$: an upper bound on the in-degree of vertex $u_{s,1}$ in H;

variables:
$a(i)\in\{0,1\}$, $i\in E_1\cup E_3$: $a(i)$ represents edge $a_i\in E_1\cup E_3$ $(a(i)=1, i\in E_1)$
$(a(i)=1\Leftrightarrow$ edge a_i is used in $H)$;

$e(s,t), e(t,s) \in \{0,1\}$, $s \in [1,s^*]$, $t \in [1,t^*]$: $e(s,t)$ (resp., $e(t,s)$) represents direction $(u_{s,1}, v_{t,1})$ (resp., $(v_{t,1}, u_{s,1})$), where $e(s,t) = 1$ (resp., $e(t,s) = 1$) $\Leftrightarrow$ edge $u_{s,1}, v_{t,1}$ is used in H and direction $(u_{s,1}, v_{t,1})$ (resp., $(v_{t,1}, u_{s,1})$) is assigned to edge $u_{s,1}, v_{t,1}$;
$\chi(t) \in [1,c^*]$, $t \in [1,t^*]$: $\chi(t)$ represents the color assigned to vertex $v_{t,1}$ ($\chi(t) = c \Leftrightarrow$ vertex $v_{t,1}$ is assigned color c);
$\mathrm{clr}(c) \in [0, n^* - s^*]$, $c \in [1,c^*]$: the number of vertices $v_{t,i}$ with color c;
$\deg^{\mathrm{co}+}(s) \in [1,4]$, $s \in [1,s^*]$: the out-degree of vertex $u_{s,1}$ in the core of H;
$\deg^{\mathrm{co}-}(s) \in [1,4]$, $s \in [1,s^*]$: the in-degree of vertex $u_{s,1}$ in the core of H;
$\delta_{\mathrm{clr}}(t,c) \in \{0,1\}$, $t \in [1,t^*]$, $c \in [1,c^*]$ ($\delta_{\mathrm{clr}}(t,c) = 1 \Leftrightarrow \chi(t) = c$);

constraints:

$$\sum_{c \in [1,c^*]} \delta_{\mathrm{clr}}(t,c) = 1, \qquad t \in [1,t^*], \tag{A10}$$

$$\sum_{c \in [1,c^*]} c \cdot \delta_{\mathrm{clr}}(t,c) = \chi(t), \qquad t \in [1,t^*], \tag{A11}$$

$$\sum_{t \in [1,t^*]} \delta_{\mathrm{clr}}(t,c) = \mathrm{clr}(c), \qquad c \in [1,c^*], \tag{A12}$$

$$e(s,t) + e(t,s) \le 1, \qquad s \in [1,s^*], t \in [1,t^*], \tag{A13}$$

$$\sum_{s \in [1,s^*] \setminus \{\mathrm{head}(c)\}} e(t,s) \le 1 - \delta_{\mathrm{clr}}(t,c), \qquad c \in [1,c^*], t \in [1,t^*], \tag{A14}$$

$$\sum_{s \in [1,s^*] \setminus \{\mathrm{tail}(c)\}} e(s,t) \le 1 - \delta_{\mathrm{clr}}(t,c), \qquad c \in [1,c^*], t \in [1,t^*], \tag{A15}$$

$$\sum_{i \in E^-_{1,3}(s)} a(i) + \sum_{t \in [1,t^*]} e(t,s) = \deg^{\mathrm{co}-}(s), \qquad s \in [1,s^*], \tag{A16}$$

$$\sum_{i \in E^+_{1,3}(s)} a(i) + \sum_{t \in [1,t^*]} e(s,t) = \deg^{\mathrm{co}+}(s), \qquad s \in [1,s^*], \tag{A17}$$

$$\underline{d}^+(s) \le \deg^{\mathrm{co}+}(s) \le \overline{d}^+(s), \qquad s \in [1,s^*], \tag{A18}$$

$$\underline{d}^-(s) \le \deg^{\mathrm{co}-}(s) \le \overline{d}^-(s), \qquad s \in [1,s^*]. \tag{A19}$$

Appendix A.3. Specification for Chemical Graphs with Rank 2

To generate any of the three rank-2 polymer topologies in $\mathcal{PT}(2,4)$, we use the scheme graph $(K = (V_K = \{u_1,u_2,u_3,u_4\}, E_K), \mathcal{E} = (E_1, E_2, E_3))$ in Figure 2d, where $s^* = |V(K)| = 4$, $c^* = |E_1 \cup E_2| = 5$, $E_1 = \{a_1 = (u_1,u_4), a_2 = (u_2,u_3), a_3 = (u_2,u_4)\}$, $E_2 = \{a_4 = (u_1,u_2), a_5 = (u_3,u_4)\}$ and $E_3 = \{a_6 = (u_1,u_2), a_7 = (u_3,u_4)\}$. Recall that each color $i \in [1,c^*]$ is assigned to edge $a_i \in E_1 \cup E_2$. We impose some more constraints on the degree of each of the vertices $u_{s,1}$, $s \in [1,s^*]$ and $v_{t,1}$, $t \in [1,t^*]$ so that the core of a selected graph H satisfies one of the three least simple graphs in Figure 2a–c. We also let a variable θ mean the topological parameter $\theta(H)$ of a selected subgraph H.

constants:
$s^* = 4$, $c^* = 5$,
$E^-_1(1) = \emptyset$, $E^-_2(1) = \emptyset$, $E^-_3(1) = \emptyset$, $E^+_1(1) = \{1\}$, $E^+_2(1) = \{4\}$, $E^+_3(1) = \{6\}$,
$E^-_1(2) = \emptyset$, $E^-_2(2) = \{4\}$, $E^-_3(2) = \{6\}$, $E^+_1(2) = \{2,3\}$, $E^+_2(2) = \emptyset$, $E^+_3(2) = \emptyset$,
$E^-_1(3) = \{2\}$, $E^-_2(3) = \emptyset$, $E^-_3(3) = \emptyset$, $E^+_1(3) = \emptyset$, $E^+_2(3) = \{5\}$, $E^+_3(3) = \{7\}$,
$E^-_1(4) = \{1,3\}$, $E^-_2(4) = \{5\}$, $E^-_3(4) = \{7\}$, $E^+_1(4) = \emptyset$, $E^+_2(4) = \emptyset$, $E^+_3(4) = \emptyset$,
$\underline{d}^-(1) = 0$, $\overline{d}^-(1) = 0$, $\underline{d}^+(1) = 2$, $\overline{d}^+(1) = 2$,
$\underline{d}^-(2) = 1$, $\overline{d}^-(2) = 2$, $\underline{d}^+(2) = 1$, $\overline{d}^+(2) = 2$,
$\underline{d}^-(3) = 1$, $\overline{d}^-(3) = 1$, $\underline{d}^+(3) = 1$, $\overline{d}^+(3) = 2$,
$\underline{d}^-(4) = 2$, $\overline{d}^-(4) = 3$, $\underline{d}^+(4) = 0$, $\overline{d}^+(4) = 0$,

variables:
$\theta \in [-n^*, n^*]$: The topology-parameter $\theta(H)$ for rank 2;

constraints:

$$a(2) + \mathrm{clr}(2) \geq 1, \tag{A20}$$
$$a(3) + \mathrm{clr}(3) + \mathrm{clr}(4) \geq 1, \tag{A21}$$
$$\mathrm{clr}(4) \geq \mathrm{clr}(5), \tag{A22}$$
$$\mathrm{clr}(3) \leq \mathrm{clr}(2) + 1, \tag{A23}$$
$$\mathrm{clr}(3) \leq \mathrm{clr}(1) + 1 + n^*(3 - \deg^{\mathrm{co}-}(4)), \tag{A24}$$
$$-\theta \leq 1 + \mathrm{clr}(2) + n^*(2 - \deg^{\mathrm{co}+}(3)), \tag{A25}$$
$$-\theta \geq 1 + \mathrm{clr}(2) - n^*(2 - \deg^{\mathrm{co}+}(3)), \tag{A26}$$
$$\theta \leq n^*(4 - \deg^{\mathrm{co}+}(2) - \deg^{\mathrm{co}-}(2)), \tag{A27}$$
$$\theta \geq -n^*(4 - \deg^{\mathrm{co}+}(2) - \deg^{\mathrm{co}-}(2)), \tag{A28}$$
$$\theta \leq 1 + \mathrm{clr}(3) + n^*(3 - \deg^{\mathrm{co}-}(4)), \tag{A29}$$
$$\theta \geq 1 + \mathrm{clr}(3) - n^*(3 - \deg^{\mathrm{co}-}(4)). \tag{A30}$$

Appendix A.4. Selecting A Subgraph

We prepare a binary variable $u(s,i)$ (resp., $v(t,i)$) for each vertex $u_{s,i}$ in tree S_s (resp., $v_{t,i}$ in tree T_t). We include constraints so that the path $(v_{1,1}, v_{1,2}, \ldots, v_{t^*,1})$ is partitioned into subpaths P_c, $c \in [1, c^*]$, where possibly some P_c is empty, and the resulting subgraph H becomes a connected rank-2 graph with $n(H) = n^*$, $\mathrm{cs}(H) = \mathrm{cs}^*$, $\mathrm{ch}(H) = \mathrm{ch}^*$ and $\theta(H) = \theta^*$.

constants:
Integers $d_{\max} \in \{3, 4\}$, $\mathrm{ch}^* \geq 0$;
Prepare the set $\mathrm{Cld}(i)$ of the indices of children of a vertex v_i
the index $\mathrm{prt}(i)$ of the parent of a non-root vertex v_i, and
the set $\mathrm{Dst}(h)$ of indices i such that the height of a vertex v_i is h
in the rooted tree $T(2, d_{\max} - 1, \mathrm{ch}^*)$;

variables:
$u(s,i) \in \{0,1\}$, $s \in [1, s^*]$, $i \in [1, n_{\mathrm{tree}}]$: $u(s,i)$ represents vertex $u_{s,i}$
($u(s,i) = 1 \Leftrightarrow$ vertex $u_{s,i}$ is used in H and edge $e'_{s,i}$ $(i \geq 2)$ is used in H);
$v(t,i) \in \{0,1\}$, $t \in [1, t^*]$, $i \in [1, n_{\mathrm{tree}}]$: $v(t,i)$ represents vertex $v_{t,i}$
($v(t,i) = 1 \Leftrightarrow$ vertex $v_{t,i}$ is used in H and edge $e_{t,i}$ $(i \geq 2)$ is used in H);
$e(t) \in \{0,1\}$, $t \in [1, t^* + 1]$: $e(t)$ represents edge $e_t = v_{t-1,1}v_{t,i}$,
where $e_{1,1}$ and $e_{t^*+1,1}$ are fictitious edges ($e(t) = 1 \Leftrightarrow$ edge e_t is used in H);

constraints:

$$u(s,1) = 1, \qquad s \in [1, s^*], \tag{A31}$$
$$d_{\max} \cdot u(t,i) \geq \sum_{j \in \mathrm{Cld}(i)} u(t,j), \qquad t \in [1, \mathrm{cs}^*], i \in [2, n_{\mathrm{in}}], \tag{A32}$$
$$v(t,1) = 1, \qquad t \in [1, t^*], \tag{A33}$$
$$d_{\max} \cdot v(t,i) \geq \sum_{j \in \mathrm{Cld}(i)} v(t,j), \qquad t \in [1, \mathrm{cs}^*], i \in [2, n_{\mathrm{in}}], \tag{A34}$$
$$\sum_{s \in [1,s^*], i \in [1, n_{\mathrm{tree}}]} u(s,i) + \sum_{t \in [1,t^*], i \in [1, n_{\mathrm{tree}}]} v(t,i) = n^*, \tag{A35}$$
$$\sum_{s \in [1,s^*], i \in \mathrm{Dst}(\mathrm{ch}^*)} u(s,i) + \sum_{t \in [1,t^*], i \in \mathrm{Dst}(\mathrm{ch}^*)} v(t,i) \geq 1, \tag{A36}$$
$$e(1) = e(t^*+1) = 0, \tag{A37}$$
$$e(t+1) + \sum_{s \in [1,s^*]} e(t,s) = 1, \qquad t \in [1, t^*], \tag{A38}$$
$$e(t) + \sum_{s \in [1,s^*]} e(s,t) = 1, \qquad t \in [1, t^*], \tag{A39}$$
$$c^* \geq \chi(1) \geq \chi(2) \geq \cdots \geq \chi(t^*) \geq 1, \tag{A40}$$
$$e(t+1) \geq 1 + \chi(t+1) - \chi(t), \qquad t \in [1, t^*-1], \tag{A41}$$
$$c^* \cdot (1 - e(t+1)) \geq \chi(t) - \chi(t+1), \qquad t \in [1, t^*-1]. \tag{A42}$$

Appendix A.5. Assigning Multiplicity

We prepare an integer variable $\widetilde{\beta}(e)$ or $\widehat{\beta}(e)$ for each edge e in the $(t^*, \mathrm{ch}^*, d_{\max})$-tree-extension of the scheme graph to denote the multiplicity of e in a selected graph H and include necessary constraints for the variables to satisfy in H.

variables:
$\widetilde{\beta}(i) \in [0,3]$, $i \in E_1 \cup E_3$: $\widetilde{\beta}(i)$ represents the multiplicity of edge a_i,
where $\widetilde{\beta}(i) = 0$ if edge a_i is not in H;
$\widetilde{\beta}(p,i) \in [0,3]$, $p \in [1, \mathrm{cs}^*]$, $i \in [2, n_{\mathrm{tree}}]$: $\widetilde{\beta}(p,i)$ with $p \leq s^*$ (resp., $p > s^*$) represents
the multiplicity of edge $e'_{p,i}$ (resp., $e_{p-s^*,i}$);
$\widetilde{\beta}(t,1) \in [0,3]$, $t \in [1, t^*+1]$: $\widetilde{\beta}(t,1)$ represents the multiplicity of edge e_t;
$\widehat{\beta}(s,t) \in [0,3]$, $s \in [1, s^*]$, $t \in [1, t^*]$: $\widehat{\beta}(s,t)$ represents the multiplicity of edge $u_{s,1}v_{t,1}$;

constraints:

$$a(i) = 1, \qquad i \in E_3, \tag{A43}$$
$$a(i) \leq \widetilde{\beta}(i) \leq 3a(i), \qquad i \in E_1 \cup E_3, \tag{A44}$$
$$u(s,i) \leq \widetilde{\beta}(s,i) \leq 3u(s,i), \qquad s \in [1, s^*], i \in [2, n_{\mathrm{tree}}], \tag{A45}$$
$$v(t,i) \leq \widetilde{\beta}(s^*+t,i) \leq 3v(t,i), \qquad t \in [1, t^*], i \in [2, n_{\mathrm{tree}}], \tag{A46}$$
$$e(t) \leq \widetilde{\beta}(t,1) \leq 3e(t), \qquad t \in [1, t^*+1], \tag{A47}$$
$$e(s,t) + e(t,s) \leq \widehat{\beta}(s,t) \leq 3e(s,t) + 3e(t,s), \qquad s \in [1, s^*], t \in [1, t^*]. \tag{A48}$$

Appendix A.6. Assigning Chemical Elements and Valence Condition

We include constraints so that each vertex v in a selected graph H satisfies the valence condition; i.e., $\sum_{uv \in E(H)} \beta(uv) \leq \mathrm{val}(\alpha(u))$. With these constraints, a rank-2 chemical graph $G = (H, \alpha, \beta)$ on a selected subgraph H will be constructed.

constants:
A set $\Lambda \cup \{\epsilon\}$ of chemical elements, where ϵ denotes null;

A coding $[\mathtt{a}]$, $\mathtt{a} \in \Lambda \cup \{\epsilon\}$ such that $[\epsilon] = 0$; $[\mathtt{a}] \geq 1$, $\mathtt{a} \in \Lambda$; and $[\mathtt{a}] \neq [\mathtt{b}]$ if $\mathtt{a} \neq \mathtt{b}$;
Let $[\Lambda]$ and $[\Lambda \cup \{\epsilon\}]$ denote $\{[\mathtt{a}] \mid \mathtt{a} \in \Lambda\}$ and $\{[\mathtt{a}] \mid \mathtt{a} \in \Lambda \cup \{\epsilon\}\}$, respectively;
A valence function: $\mathrm{val} : \Lambda \to [1, 4]$;

variables:
$\widetilde{\alpha}(p,i) \in [\Lambda \cup \{\epsilon\}]$, $p \in [1, \mathrm{cs}^*]$, $i \in [1, n_{\mathrm{tree}}]$:
$\widetilde{\alpha}(p,i)$ with $p \leq s^*$ (resp., $p > s^*$) represents $\alpha(u_{p,i})$ (resp., $\alpha(v_{p-s^*,i})$);
$\delta_\alpha(p,i,\mathtt{a}) \in \{0,1\}$, $p \in [1, \mathrm{cs}^*]$, $i \in [1, n_{\mathrm{tree}}]$, $\mathtt{a} \in \Lambda \cup \{\epsilon\}$:
$\delta_\alpha(p,i,\mathtt{a}) = 1 \Leftrightarrow \alpha(u_{p,i}) = \mathtt{a}$ for $p \leq s^*$ and $\alpha(v_{p-s^*,i}) = \mathtt{a}$ for $p > s^*$;
$\delta_{\widetilde{\beta}}(i,k) \in \{0,1\}$, $p \in [1, \mathrm{cs}^*]$, $i \in E_1 \cup E_3$, $k \in [0,3]$:
$\delta_{\widetilde{\beta}}(i,k) = 1 \Leftrightarrow$ the multiplicity of edge a_i in H is k;
$\delta_{\widetilde{\beta}}(p,i,k) \in \{0,1\}$, $p \in [1, \mathrm{cs}^*]$, $i \in [2, n_{\mathrm{tree}}]$, $k \in [0,3]$:
$\delta_{\widetilde{\beta}}(p,i,k) = 1 \Leftrightarrow$ the multiplicity of edge $e'_{p,i}$, $p \leq s^*$ (or $e_{p-s^*,i}$, $p > s^*$) in H is k;
$\delta_{\widetilde{\beta}}(t,1,k) \in \{0,1\}$, $t \in [1, t^* + 1]$, $k \in [0,3]$:
$\delta_{\widetilde{\beta}}(t,1,k) = 1 \Leftrightarrow$ the multiplicity of edge e_t in H is k;
$\delta_{\widehat{\beta}}(s,t,k) \in \{0,1\}$, $s \in [1, s^*]$, $t \in [1, t^*]$, $k \in [0,3]$:
$\delta_{\widehat{\beta}}(s,t,k) = 1 \Leftrightarrow$ the multiplicity of edge $u_{s,1}v_{t,1}$ in H is k;

constraints:

$$\sum_{\mathtt{a} \in \Lambda \cup \{\epsilon\}} \delta_\alpha(p,i,\mathtt{a}) = 1, \qquad p \in [1, \mathrm{cs}^*], i \in [1, n_{\mathrm{tree}}], \tag{A49}$$

$$\sum_{\mathtt{a} \in \Lambda \cup \{\epsilon\}} [\mathtt{a}] \cdot \delta_\alpha(p,i,\mathtt{a}) = \widetilde{\alpha}(p,i), \qquad p \in [1, \mathrm{cs}^*], i \in [1, n_{\mathrm{tree}}], \tag{A50}$$

$$\sum_{k \in [0,3]} \delta_{\widetilde{\beta}}(i,k) = 1, \qquad i \in E_1 \cup E_3, \tag{A51}$$

$$\sum_{k \in [1,3]} k \cdot \delta_{\widetilde{\beta}}(i,k) = \widetilde{\beta}(i), \qquad i \in E_1 \cup E_3, \tag{A52}$$

$$\sum_{k \in [0,3]} \delta_{\widetilde{\beta}}(p,i,k) = 1, \qquad p \in [1, \mathrm{cs}^*], i \in [2, n_{\mathrm{tree}}], \tag{A53}$$

$$\sum_{k \in [1,3]} k \cdot \delta_{\widetilde{\beta}}(p,i,k) = \widetilde{\beta}(p,i), \qquad p \in [1, \mathrm{cs}^*], i \in [2, n_{\mathrm{tree}}], \tag{A54}$$

$$\sum_{k \in [0,3]} \delta_{\widetilde{\beta}}(t,1,k) = 1, \qquad t \in [1, t^* + 1], \tag{A55}$$

$$\sum_{k \in [1,3]} k \cdot \delta_{\widetilde{\beta}}(t,1,k) = \widetilde{\beta}(t,1), \qquad t \in [1, t^* + 1], \tag{A56}$$

$$\sum_{k \in [0,3]} \delta_{\widehat{\beta}}(s,t,k) = 1, \qquad s \in [1, s^*], t \in [1, t^*], \tag{A57}$$

$$\sum_{k \in [0,3]} k\delta_{\widehat{\beta}}(s,t,k) = \widehat{\beta}(s,t), \qquad s \in [1, s^*], t \in [1, t^*], \tag{A58}$$

$$\sum_{i\in E_{1,3}(s)} \widetilde{\beta}(i) + \sum_{t\in[1,t^*]} \widehat{\beta}(s,t) + \sum_{j\in \mathrm{Cld}(1)} \widetilde{\beta}(s,j) \le \sum_{\mathtt{a}\in\Lambda} \mathrm{val}(\mathtt{a})\cdot \delta_\alpha(s,1,\mathtt{a}), \qquad s\in[1,s^*], \tag{A59}$$

$$\sum_{s\in[1,s^*]} \widehat{\beta}(s,t) + \widetilde{\beta}(t,1) + \widetilde{\beta}(t+1,1) + \sum_{j\in \mathrm{Cld}(1)} \widetilde{\beta}(s^*+t,j) \le \sum_{\mathtt{a}\in\Lambda} \mathrm{val}(\mathtt{a})\cdot \delta_\alpha(s^*+t,1,\mathtt{a}), \qquad t\in[1,t^*], \tag{A60}$$

$$\widetilde{\beta}(p,i) + \sum_{j\in \mathrm{Cld}(i)} \widetilde{\beta}(p,j) \le \sum_{\mathtt{a}\in\Lambda} \mathrm{val}(\mathtt{a})\cdot \delta_\alpha(p,i,\mathtt{a}), \qquad p\in[1,\mathrm{cs}^*], i\in[2,n_{\mathrm{tree}}]. \tag{A61}$$

Appendix A.7. Descriptors for Mass, the Numbers of Elements and Bonds

We include constraints to compute descriptors $\overline{\mathrm{ms}}(G)$ $\mathrm{ce}^{\mathrm{co}}_{\mathtt{a}}(G)$, $\mathrm{ce}^{\mathrm{nc}}_{\mathtt{a}}(G)$ ($\mathtt{a}\in\Lambda$), $b_k(G)$ ($k\in[2,3]$) and $n_{\mathtt{H}}(G)$ according to the definitions in Section 2.1.2.

constants:
A function $\mathrm{mass}^*:\Lambda\to\mathbb{Z}$; Let $\mathrm{mass}(\mathtt{a})$ denote the observed mass of a chemical element $\mathtt{a}\in\Lambda$, and define $\mathrm{mass}^*(\mathtt{a}) = \lfloor 10\cdot \mathrm{mass}(\mathtt{a})\rfloor$;

variables:
$\mathrm{ce}^{\mathrm{co}}(\mathtt{a})\in[0,n^*]$, $\mathtt{a}\in\Lambda$;
$\mathrm{ce}^{\mathrm{nc}}(\mathtt{a})\in[0,n^*]$, $\mathtt{a}\in\Lambda$;
$\mathrm{Mass}\in\mathbb{Z}$;
$\mathrm{b}^{\mathrm{co}}(k)\in[0,2n^*]$, $k\in[1,3]$;
$\mathrm{b}^{\mathrm{nc}}(k)\in[0,2n^*]$, $k\in[1,3]$;
$\mathrm{n}_{\mathtt{H}}\in[0,4n^*]$: the number of hydrogen atoms to be included in G;

constraints:

$$\sum_{p\in[1,\mathrm{cs}^*]} \delta_\alpha(p,1,\mathtt{a}) = \mathrm{ce}^{\mathrm{co}}(\mathtt{a}), \qquad \mathtt{a}\in\Lambda, \tag{A62}$$

$$\sum_{p\in[1,\mathrm{cs}^*], i\in[2,n_{\mathrm{tree}}]} \delta_\alpha(p,i,\mathtt{a}) = \mathrm{ce}^{\mathrm{nc}}(\mathtt{a}), \qquad \mathtt{a}\in\Lambda, \tag{A63}$$

$$\sum_{\mathtt{a}\in\Lambda} \mathrm{mass}^*(\mathtt{a})(\mathrm{ce}^{\mathrm{co}}(\mathtt{a}) + \mathrm{ce}^{\mathrm{nc}}(\mathtt{a})) = \mathrm{Mass}, \tag{A64}$$

$$\sum_{i\in E_1\cup E_3} \delta_{\widetilde{\beta}}(i,k) + \sum_{s\in[1,s^*],t\in[1,t^*]} \delta_{\widehat{\beta}}(s,t,k) + \sum_{t\in[2,t^*]} \delta_{\widetilde{\beta}}(t,1,k) = \mathrm{b}^{\mathrm{co}}(k), \qquad k\in[1,3], \tag{A65}$$

$$\sum_{p\in[1,\mathrm{cs}^*], i\in[2,n_{\mathrm{tree}}]} \delta_{\widetilde{\beta}}(p,i,k) = \mathrm{b}^{\mathrm{nc}}(k), \qquad k\in[1,3], \tag{A66}$$

$$\sum_{\mathtt{a}\in\Lambda} \mathrm{val}(\mathtt{a})(\mathrm{ce}^{\mathrm{co}}(\mathtt{a}) + \mathrm{ce}^{\mathrm{nc}}(\mathtt{a})) - 2(n^* + 1 + \mathrm{b}^{\mathrm{co}}(2) + \mathrm{b}^{\mathrm{nc}}(2) + 2\mathrm{b}^{\mathrm{co}}(3) + 2\mathrm{b}^{\mathrm{nc}}(3)) = \mathrm{n}_{\mathtt{H}}. \tag{A67}$$

Appendix A.8. Descriptor for the Number of Specified Degree

We include constraints to compute descriptors $\mathrm{dg}_i(G)$ ($i\in[1,4]$) according to the definitions in Section 2.1.2. We also add constraints so that the maximum degree of a non-core vertex in H is at most 3 (resp., equal to 4) when $d_{\max}=3$ (resp., $d_{\max}=4$).

variables:
$\deg(p,i) \in [0,4]$, $p \in [1, \mathrm{cs}^*]$, $i \in [1, n_{\mathrm{tree}}]$:
$\deg(p,i)$ represents $\deg_H(u_{p,i})$ for $p \leq s^*$ or $\deg_H(v_{p-s^*,i})$ for $p > s^*$;
$\delta_{\deg}(p,i,d) \in \{0,1\}$, $p \in [1, \mathrm{cs}^*]$, $i \in [1, n_{\mathrm{tree}}]$, $d \in [0,4]$:
$\delta_{\deg}(p,i,d) = 1 \Leftrightarrow \deg(p,i) = d$;
$\mathrm{dg}(d) \in [0, n^*]$, $d \in [1,4]$;

constraints:

$$\sum_{i \in E_{1,3}(s)} a(i) + \sum_{t \in [1,t^*]} (e(s,t) + e(t,s)) + \sum_{j \in \mathrm{Cld}(1)} u(s,j) = \deg(s,1), \qquad s \in [1, s^*], \tag{A68}$$

$$u(s,i) + \sum_{j \in \mathrm{Cld}(i)} u(s,j) = \deg(s,i), \qquad s \in [1,s^*], i \in [2, n_{\mathrm{tree}}], \tag{A69}$$

$$2 + \sum_{j \in \mathrm{Cld}(1)} v(t,j) = \deg(s^*+t,1), \qquad t \in [1,t^*], \tag{A70}$$

$$v(t,i) + \sum_{j \in \mathrm{Cld}(i)} v(t,j) = \deg(s^*+t,i), \qquad t \in [1,t^*], i \in [2, n_{\mathrm{tree}}], \tag{A71}$$

$$\sum_{d \in [0,4]} \delta_{\deg}(p,i,d) = 1, \qquad p \in [1, \mathrm{cs}^*], i \in [1, n_{\mathrm{tree}}], \tag{A72}$$

$$\sum_{d \in [1,4]} d \cdot \delta_{\deg}(p,i,d) = \deg(p,i), \qquad p \in [1, \mathrm{cs}^*], i \in [1, n_{\mathrm{tree}}], \tag{A73}$$

$$\sum_{p \in [1, \mathrm{cs}^*], i \in [1, n_{\mathrm{tree}}]} \delta_{\deg}(p,i,d) = \mathrm{dg}(d), \qquad d \in [1,4], \tag{A74}$$

$$\sum_{p \in [1, \mathrm{cs}^*], i \in [2, n_{\mathrm{tree}}]} \delta_{\deg}(p,i,4) \geq 1 \text{ (resp., } = 0) \qquad \text{when } d_{\max} = 4 \text{ (resp., } = 3). \tag{A75}$$

Appendix A.9. Descriptor for the Number of Adjacency-Configurations

We include constraints to compute descriptors $\mathrm{ac}^{\mathrm{co}}_\gamma(G)$ and $\mathrm{ac}^{\mathrm{nc}}_\gamma(G)$ ($\gamma = (\mathtt{a}, \mathtt{b}, k) \in \Gamma$) according to the definitions in Section 2.1.2.

constants:
A set $\Gamma = \Gamma_< \cup \Gamma_= \cup \Gamma_>$ of proper tuples $(\mathtt{a}, \mathtt{b}, k) \in \Lambda \times \Lambda \times [1,3]$;
The set $\Gamma_0 = \{(\mathtt{a}, \mathtt{b}, 0) \mid \mathtt{a}, \mathtt{b} \in \Lambda \cup \{\epsilon\}\}$;

variables:
$\delta_\tau(i,\gamma) \in \{0,1\}$, $i \in E_1 \cup E_3$, $\gamma \in \Gamma \cup \Gamma_0$:
$\delta_\tau(i,\gamma) = 1 \Leftrightarrow$ edge a_i is assigned tuple γ; i.e., $\gamma = (\widetilde{\alpha}(\mathrm{tail}(i),1), \widetilde{\alpha}(\mathrm{head}(i),1), \widetilde{\beta}(i))$;
$\delta_\tau(t,1,\gamma) \in \{0,1\}$, $t \in [2,t^*]$, $\gamma \in \Gamma \cup \Gamma_0$:
$\delta_\tau(t,1,\gamma) = 1 \Leftrightarrow$ edge e_t is assigned tuple γ; i.e., $\gamma = (\widetilde{\alpha}(s^*+t-1,1), \widetilde{\alpha}(s^*+t,1), \widetilde{\beta}(t,1))$;
$\delta_\tau(t,i,\gamma) \in \{0,1\}$, $p \in [1, \mathrm{cs}^*]$, $i \in [2, n_{\mathrm{tree}}]$, $\gamma \in \Gamma \cup \Gamma_0$:
$\delta_\tau(t,i,\gamma) = 1 \Leftrightarrow$ edge $e'_{p,i}$, $p \leq s^*$ (or $e_{p-s^*,i}$, $p > s^*$) is assigned tuple γ; i.e.,
$\gamma = (\widetilde{\alpha}(p, \mathrm{prt}(i)), \widetilde{\alpha}(p,i), \widetilde{\beta}(p,i))$;
$\delta_{\widetilde{\tau}}(s,t,\gamma) \in \{0,1\}$, $s \in [1,s^*]$, $t \in [1,t^*]$, $\gamma \in \Gamma \cup \Gamma_0$:
$\delta_{\widetilde{\tau}}(s,t,\gamma) = 1 \Leftrightarrow$ edge $u_{s,1}v_{t,1}$ is assigned tuple γ; i.e., $\gamma = (\widetilde{\alpha}(s,1), \widetilde{\alpha}(s^*+t,1), \widehat{\beta}(s,t))$;
$\mathrm{ac}^{\mathrm{co}}(\gamma) \in [0, n^*]$, $\gamma \in \Gamma_< \cup \Gamma_=$;
$\mathrm{ac}^{\mathrm{nc}}(\gamma) \in [0, n^*]$, $\gamma \in \Gamma_< \cup \Gamma_=$;

constraints:

$$\sum_{\gamma\in\Gamma\cup\Gamma_0}\delta_\tau(i,\gamma)=1, \qquad i\in E_1\cup E_3, \tag{A76}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}[\mathtt{a}]\delta_\tau(i,(\mathtt{a},\mathtt{b},k))=\widetilde{\alpha}(\mathrm{tail}(i),1), \qquad i\in E_1\cup E_3, \tag{A77}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}[\mathtt{b}]\delta_\tau(i,(\mathtt{a},\mathtt{b},k))=\widetilde{\alpha}(\mathrm{head}(i),1), \qquad i\in E_1\cup E_3, \tag{A78}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}k\cdot\delta_\tau(i,(\mathtt{a},\mathtt{b},k))=\widetilde{\beta}(i), \qquad i\in E_1\cup E_3, \tag{A79}$$

$$\sum_{\gamma\in\Gamma\cup\Gamma_0}\delta_\tau(t,1,\gamma)=1, \qquad t\in[2,t^*], \tag{A80}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}[\mathtt{a}]\delta_\tau(t,1,(\mathtt{a},\mathtt{b},k))=\widetilde{\alpha}(s^*+t-1,1), \qquad t\in[2,t^*], \tag{A81}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}[\mathtt{b}]\delta_\tau(t,1,(\mathtt{a},\mathtt{b},k))=\widetilde{\alpha}(s^*+t,1), \qquad t\in[2,t^*], \tag{A82}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}k\cdot\delta_\tau(t,1,(\mathtt{a},\mathtt{b},k))=\widetilde{\beta}(t,1), \qquad t\in[2,t^*], \tag{A83}$$

$$\sum_{\gamma\in\Gamma\cup\Gamma_0}\delta_\tau(p,i,\gamma)=1, \qquad p\in[1,\mathrm{cs}^*],i\in[2,n_{\mathrm{tree}}], \tag{A84}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}[\mathtt{a}]\delta_\tau(p,i,(\mathtt{a},\mathtt{b},k))=\widetilde{\alpha}(p,\mathrm{prt}(i)), \qquad p\in[1,\mathrm{cs}^*],i\in[2,n_{\mathrm{tree}}], \tag{A85}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}[\mathtt{b}]\delta_\tau(p,i,(\mathtt{a},\mathtt{b},k))=\widetilde{\alpha}(p,i), \qquad p\in[1,\mathrm{cs}^*],i\in[2,n_{\mathrm{tree}}], \tag{A86}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}k\cdot\delta_\tau(p,i,(\mathtt{a},\mathtt{b},k))=\widetilde{\beta}(p,i), \qquad p\in[1,\mathrm{cs}^*],i\in[2,n_{\mathrm{tree}}], \tag{A87}$$

$$\sum_{\gamma\in\Gamma\cup\Gamma_0}\delta_{\widehat{\tau}}(s,t,\gamma)=1, \qquad s\in[1,s^*],t\in[1,t^*], \tag{A88}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}[\mathtt{a}]\delta_{\widehat{\tau}}(s,t,(\mathtt{a},\mathtt{b},k))=\widetilde{\alpha}(s,1), \qquad s\in[1,s^*],t\in[1,t^*], \tag{A89}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}[\mathtt{b}]\delta_{\widehat{\tau}}(s,t,(\mathtt{a},\mathtt{b},k))=\widetilde{\alpha}(s^*+t,1), \qquad s\in[1,s^*],t\in[1,t^*], \tag{A90}$$

$$\sum_{(\mathtt{a},\mathtt{b},k)\in\Gamma\cup\Gamma_0}k\cdot\delta_{\widehat{\tau}}(s,t,(\mathtt{a},\mathtt{b},k))=\widehat{\beta}(s,t), \qquad s\in[1,s^*],t\in[1,t^*], \tag{A91}$$

$$\begin{aligned}&\sum_{i\in E_1\cup E_3}(\delta_\tau(i,\gamma)+\delta_\tau(i,\overline{\gamma}))\\&+\sum_{s\in[1,s^*],t\in[1,t^*]}(\delta_{\widehat{\tau}}(s,t,\gamma)+\delta_{\widehat{\tau}}(s,t,\overline{\gamma}))\\&+\sum_{t\in[2,t^*]}(\delta_\tau(t,1,\gamma)+\delta_\tau(t,1,\overline{\gamma}))=\mathrm{ac}^{\mathrm{co}}(\gamma),\end{aligned} \qquad \gamma\in\Gamma_<, \tag{A92}$$

$$\begin{aligned}&\sum_{i\in E_1\cup E_3}\delta_\tau(i,\gamma)+\sum_{s\in[1,s^*],t\in[1,t^*]}\delta_{\widehat{\tau}}(s,t,\gamma)\\&+\sum_{t\in[2,t^*]}\delta_\tau(t,1,\gamma)=\mathrm{ac}^{\mathrm{co}}(\gamma),\end{aligned} \qquad \gamma\in\Gamma_=, \tag{A93}$$

$$\sum_{p\in[1,\mathrm{cs}^*],i\in[2,n_{\mathrm{tree}}]}(\delta_\tau(p,i,\gamma)+\delta_\tau(p,i,\overline{\gamma}))=\mathrm{ac}^{\mathrm{nc}}(\gamma), \qquad \gamma\in\Gamma_<, \tag{A94}$$

$$\sum_{p\in[1,\mathrm{cs}^*],i\in[2,n_{\mathrm{tree}}]}\delta_\tau(p,i,\gamma)=\mathrm{ac}^{\mathrm{nc}}(\gamma), \qquad \gamma\in\Gamma_=. \tag{A95}$$

Appendix A.10. Descriptor for 1-Path Connectivity

We include constraints to compute descriptor $\kappa_1(G)$ according to the definition.

variables:

A real variable $\kappa_1 \geq 0$;
$\delta_{\mathrm{dd}}(i,d,d',\mu) \in \{0,1\}$, $i \in E_1 \cup E_3$, $d,d' \in [0,4]$, $\mu \in \{0,1\}$:
$\delta_{\mathrm{dd}}(i,d,d',\mu) = 1 \Leftrightarrow \deg_H(u_{\mathrm{tail}}(i)) = d$ and $\deg_H(u_{\mathrm{head}}(i)) = d'$,
where a_i is in H if and only if $\mu = 1$;
$\delta_{\mathrm{dd}}(t,1,d,d',\mu) \in \{0,1\}$, $t \in [2,t^*]$, $d,d' \in [0,4]$: $\delta_{\mathrm{dd}}(t,1,d,d',\mu) = 1 \Leftrightarrow$
$\deg_H(v_{t-1,1}) = d$ and $\deg_H(v_{t,1}) = d'$ where e_t is in H if and only if $\mu = 1$;
$\delta_{\mathrm{dd}}(p,i,d,d',\mu) \in \{0,1\}$, $p \in [1,\mathrm{cs}^*]$, $i \in [2,n_{\mathrm{tree}}]$, $d,d' \in [0,4]$: $\delta_{\mathrm{dd}}(p,i,d,d',\mu) = 1 \Leftrightarrow$
$\deg_H(u_{p,\mathrm{prt}(i)}) = d$ and $\deg_H(u_{p,i}) = d'$ for $p \leq s^*$
(or $\deg_H(v_{p-s^*,\mathrm{prt}(i)}) = d$ and $\deg_H(v_{p-s^*,i}) = d'$ for $p > s^*$),
where edge $e'_{p,i}$ or $e_{p-s^*,i}$ is in H if and only if $\mu = 1$;
$\delta_{\widehat{\mathrm{dd}}}(s,t,d,d',\mu) \in \{0,1\}$, $s \in [1,s^*]$, $t \in [1,t^*]$, $d,d' \in [0,4]$, $\mu \in \{0,1\}$:
$\delta_{\widehat{\mathrm{dd}}}(s,t,d,d',1) = 1 \Leftrightarrow \deg_H(u_{s,1}) = d$ and $\deg_H(v_{t,1}) = d'$,
where $u_{s,1}v_{t,1}$ is in H if and only if $\mu = 1$;

constraints:

$$\sum_{d,d' \in [0,4], \mu \in \{0,1\}} \delta_{\mathrm{dd}}(i,d,d',\mu) = 1, \qquad i \in E_1 \cup E_3, \tag{A96}$$

$$\sum_{d,d' \in [0,4], \mu \in \{0,1\}} \mu \cdot \delta_{\mathrm{dd}}(i,d,d',\mu) = a(i), \qquad i \in E_1 \cup E_3, \tag{A97}$$

$$\sum_{d \in [1,4], d' \in [0,4], \mu \in \{0,1\}} d \cdot \delta_{\mathrm{dd}}(i,d,d',\mu) = \deg(\mathrm{tail}(i),1), \qquad i \in E_1 \cup E_3, \tag{A98}$$

$$\sum_{d \in [0,4], d' \in [1,4], \mu \in \{0,1\}} d' \cdot \delta_{\mathrm{dd}}(i,d,d',\mu) = \deg(\mathrm{head}(i),1), \qquad i \in E_1 \cup E_3, \tag{A99}$$

$$\sum_{d,d' \in [0,4], \mu \in \{0,1\}} \delta_{\mathrm{dd}}(t,1,d,d',\mu) = 1, \qquad t \in [2,t^*], \tag{A100}$$

$$\sum_{d,d' \in [0,4], \mu \in \{0,1\}} \mu \cdot \delta_{\mathrm{dd}}(t,1,d,d',\mu) = e(t), \qquad t \in [2,t^*], \tag{A101}$$

$$\sum_{d \in [1,4], d' \in [0,4], \mu \in \{0,1\}} d \cdot \delta_{\mathrm{dd}}(t,1,d,d',\mu) = \deg(s^*+t-1,1), \qquad t \in [2,t^*], \tag{A102}$$

$$\sum_{d \in [0,4], d' \in [1,4], \mu \in \{0,1\}} d' \cdot \delta_{\mathrm{dd}}(t,1,d,d',\mu) = \deg(s^*+t,1), \qquad t \in [2,t^*], \tag{A103}$$

$$\sum_{d,d' \in [0,4], \mu \in \{0,1\}} \delta_{\mathrm{dd}}(p,i,d,d',\mu) = 1, \qquad p \in [1,\mathrm{cs}^*], i \in [2,n_{\mathrm{tree}}], \tag{A104}$$

$$\sum_{d,d' \in [0,4], \mu \in \{0,1\}} \mu \cdot \delta_{\mathrm{dd}}(s,i,d,d',\mu) = u(s,i), \qquad s \in [1,s^*], i \in [2,n_{\mathrm{tree}}], \tag{A105}$$

$$\sum_{d,d' \in [0,4], \mu \in \{0,1\}} \mu \cdot \delta_{\mathrm{dd}}(s^*+t,i,d,d',\mu) = v(t,i), \qquad t \in [1,t^*], i \in [2,n_{\mathrm{tree}}], \tag{A106}$$

$$\sum_{d \in [1,4], d' \in [0,4], \mu \in \{0,1\}} d \cdot \delta_{\mathrm{dd}}(p,i,d,d',\mu) = \deg(p,\mathrm{prt}(i)), \qquad p \in [1,\mathrm{cs}^*], i \in [2,n_{\mathrm{tree}}], \tag{A107}$$

$$\sum_{d \in [0,4], d' \in [1,4], \mu \in \{0,1\}} d' \cdot \delta_{\mathrm{dd}}(t,i,d,d',\mu) = \deg(p,i), \qquad p \in [1,\mathrm{cs}^*], i \in [2,n_{\mathrm{tree}}], \tag{A108}$$

$$\sum_{d,d' \in [1,4], \mu \in \{0,1\}} \delta_{\widehat{\mathrm{dd}}}(s,t,d,d',\mu) = 1, \qquad s \in [1,s^*], t \in [1,t^*], \tag{A109}$$

$$\sum_{d,d'\in[1,4],\mu\in\{0,1\}} \mu\cdot\delta_{\widehat{\mathrm{dd}}}(s,t,d,d',\mu) = e(s,t)+e(t,s), \qquad s\in[1,s^*],t\in[1,t^*], \tag{A110}$$

$$\sum_{d\in[1,4],d'\in[0,4],\mu\in\{0,1\}} d\cdot\delta_{\widehat{\mathrm{dd}}}(s,t,d,d',\mu) = \deg(s,1), \qquad s\in[1,s^*],t\in[1,t^*], \tag{A111}$$

$$\sum_{d\in[0,4],d'\in[1,4],\mu\in\{0,1\}} d'\cdot\delta_{\widehat{\mathrm{dd}}}(s,t,d,d',\mu) = \deg(s^*+t,1), \qquad s\in[1,s^*],t\in[1,t^*], \tag{A112}$$

$$\begin{aligned}(1-\xi)\kappa_1 \le &\sum_{i\in E_1\cup E_3,d,d'\in[1,4]} \delta_{\mathrm{dd}}(i,d,d',1)/\sqrt{dd'}\\ &+\sum_{t\in[2,t^*],d,d'\in[1,4]} \delta_{\mathrm{dd}}(t,1,d,d',1)/\sqrt{dd'}\\ &+\sum_{\substack{p\in[1,\mathrm{cs}^*],i\in[2,n_{\mathrm{tree}}],\\ d,d'\in[1,4]}} \delta_{\mathrm{dd}}(p,i,d,d',1)/\sqrt{dd'}\\ &+\sum_{\substack{s\in[1,s^*],t\in[1,t^*],\\ d,d'\in[1,4]}} \delta_{\widehat{\mathrm{dd}}}(s,t,d,d',1)/\sqrt{dd'} \le (1+\xi)\kappa_1,\end{aligned} \tag{A113}$$

where a tolerance ξ is set to be 0.001.

Appendix A.11. Constraints for Left-Heavy Trees

To reduce the number of rank-2 chemical graphs G that are isomorphic to each other, we include in $\mathcal{C}_2$ some additional constraints so that each subtree T' selected from tree S_p or T_t satisfies the following property:

> for any two siblings $u(p,j_1)$ and $u(p,j_2)$, $j_1<j_2$ in T', the number of descendants of $u(p,j_1)$ is not smaller than that of $u(p,j_2)$.

For this, we define $\mathrm{dsn}(p,i)$ to be the number of descendants of a vertex $u_{p,i}$ (or $v_{p-s^*,i}$) in a selected graph H and $\eta(p,i) \triangleq 21|\Lambda|\mathrm{dsn}(p,i)+20\widetilde{\alpha}(p,i)+4\deg(p,i)+\widetilde{\beta}(p,i)$, $p\in[1,\mathrm{cs}^*]$, $i\in[2,n_{\mathrm{tree}}]$. We include constraints that compute the values of dsn recursively.

variables:
$\mathrm{dsn}(p,i)\in[1,n_{\mathrm{tree}}]$, $p\in[1,\mathrm{cs}^*]$, $i\in[1,n_{\mathrm{tree}}]$: the number of descendants of vertex $u_{p,i}$ in tree S_p for $p\le s^*$ and vertex $v_{p-s^*,i}$ in tree T_{p-s^*} for $p>s^*$;
constraints:

$$\mathrm{dsn}(s,i) \ge \sum_{j\in\mathrm{Cld}(i)} \mathrm{dsn}(s,j)+u(s,i), \qquad s\in[1,s^*],i\in[1,n_{\mathrm{tree}}], \tag{A114}$$

$$\mathrm{dsn}(s^*+t,i) \ge \sum_{j\in\mathrm{Cld}(i)} \mathrm{dsn}(s^*+t,j)+v(t,i), \qquad t\in[s^*+1,\mathrm{cs}^*],i\in[1,n_{\mathrm{tree}}], \tag{A115}$$

$$\sum_{p\in[1,\mathrm{cs}^*]} \mathrm{dsn}(p,1) \le n^*, \tag{A116}$$

$$\eta(p,j_1) \ge \eta(p,j_2), \qquad p\in[1,\mathrm{cs}^*],j_1,j_2\in\mathrm{Cld}(1),j_1<j_2, \tag{A117}$$

$$\eta(p,j_1) \ge \eta(p,j_2), \qquad p\in[1,\mathrm{cs}^*],i\in[2,n_{\mathrm{in}}],j_1,j_2\in\mathrm{Cld}(i),\ j_1<j_2,\ \text{for } d_{\max}=3, \tag{A118}$$

$$\eta(p,j_1) \ge \eta(p,j_2) \ge \eta(p,j_3), \qquad p\in[1,\mathrm{cs}^*],i\in[2,n_{\mathrm{in}}],j_1,j_2,j_3\in\mathrm{Cld}(i),\ j_1<j_2<j_3,\ \text{for } d_{\max}=4. \tag{A119}$$

References

1. Miyao, T.; Kaneko, H.; Funatsu, K. Inverse QSPR/QSAR analysis for chemical structure generation (from y to x). *J. Chem. Inf. Model.* **2016**, *56*, 286–299. [CrossRef] [PubMed]
2. Skvortsova, M.I.; Baskin, I.I.; Slovokhotova, O.L.; Palyulin, V.A.; Zefirov, N.S. Inverse problem in QSAR/QSPR studies for the case of topological indices characterizing molecular shape (Kier indices). *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 630–634. [CrossRef]
3. Ikebata, H.; Hongo, K.; Isomura, T.; Maezono, R.; Yoshida, R. Bayesian molecular design with a chemical language model. *J. Comput. Aided Mol. Des.* **2017**, *31*, 379–391. [CrossRef] [PubMed]
4. Rupakheti, C.; Virshup, A.; Yang, W.; Beratan, D.N. Strategy to discover diverse optimal molecules in the small molecule universe. *J. Chem. Inf. Model.* **2015**, *55*, 529–537. [CrossRef] [PubMed]
5. Fujiwara, H.; Wang, J.; Zhao, L.; Nagamochi, H.; Akutsu, T. Enumerating treelike chemical graphs with given path frequency. *J. Chem. Inf. Model.* **2008**, *48*, 1345–1357. [CrossRef] [PubMed]
6. Kerber, A.; Laue, R.; Grüner, T.; Meringer, M. MOLGEN 4.0. *Match Commun. Math. Comput. Chem.* **1998**, *37*, 205–208.
7. Li, J.; Nagamochi, H.; Akutsu, T. Enumerating substituted benzene isomers of tree-like chemical graphs. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2016**, *15*, 633–646. [CrossRef] [PubMed]
8. Reymond, J.L. The chemical space project. *Accounts Chem. Res.* **2015**, *48*, 722–730. [CrossRef] [PubMed]
9. Akutsu, T.; Fukagawa, D.; Jansson, J.; Sadakane, K. Inferring a Graph From Path Frequency. *Discret. Appl. Math.* **2012**, *160*, 1416–1428. [CrossRef]
10. Nagamochi, H. A detachment algorithm for inferring a graph from path frequency. *Algorithmica* **2009**, *53*, 207–224. [CrossRef]
11. Fazekas, S.Z.; Ito, H.; Okuno, Y.; Seki, S.; Taneishi, K. On computational complexity of graph inference from counting. *Nat. Comput.* **2013**, *12*, 589–603. [CrossRef]
12. Bohacek, R.S.; McMartin, C.; Guida, W.C. The art and practice of structure-based drug design: A molecular modeling perspective. *Med. Res. Rev.* **1996**, *16*, 3–50. [CrossRef]
13. Gómez-Bombarelli, R.; Wei, J.N.; Duvenaud, D.; Hernández-Lobato, J.M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T.D.; Adams, R.P.; Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **2018**, *4*, 268–276. [CrossRef] [PubMed]
14. Segler, M.H.S.; Kogej, T.; Tyrchan, C.; Waller, M.P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **2017**, *4*, 120–131. [CrossRef] [PubMed]
15. Yang, X.; Zhang, J.; Yoshizoe, K.; Terayama, K.; Tsuda, K. ChemTS: an efficient python library for de novo molecular generation. *Sci. Technol. Adv. Mater.* **2017**, *18*, 972–976. [CrossRef] [PubMed]
16. Kusner, M.J.; Paige, B.; Hernández-Lobato, J.M. Grammar variational autoencoder. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1945–1954.
17. Akutsu, T.; Nagamochi, H. A Mixed Integer Linear Programming Formulation to Artificial Neural Networks. In Proceedings of the 2nd International Conference on Information Science and Systems, Tokyo, Japan, 16–19 March 2019; pp. 215–220.
18. Azam, N.A.; 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. In Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies, Valletta, Malta, 24–26 February 2020; Volume 3, pp. 101–108.
19. Chiewvanichakorn, R.; Wang, C.; Zhang, Z.; Shurbevski, A.; Nagamochi, H.; Akutsu, T. A method for the inverse QSAR/QSPR based on artificial neural networks and mixed integer linear programming. In Proceedings of the ICBBB2020, Kyoto, Japan, 19–22 January 2020.
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. In Proceedings of the 33rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Kitakyushu, Japan, 22–25 September 2020.

21. Ito, R.; Azam, N.A.; 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, 2020. In Proceedings of the BIOCOMP 2020, Las Vegas, NV, USA, 27–30 July 2020.
22. Suzuki, M.; Nagamochi, H.; Akutsu, T. Efficient enumeration of monocyclic chemical graphs with given path frequencies. *J. Cheminform.* **2014**, *6*, 31. [CrossRef] [PubMed]
23. Tezuka, Y.; Oike, H. Topological polymer chemistry. *Prog. Polym. Sci.* **2002**, *27*, 1069–1122. [CrossRef]
24. Netzeva, T.I.; Worth, A.P.; Aldenberg, T.; Benigni, R.; Cronin, M.T.; Gramatica, P.; Jaworska, J.S.; Kahn, S.; Klopman, G.; Marchant, C.A.; et al. Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships: The report and recommendations of ECVAM workshop 52. *Altern. Lab. Anim.* **2005**, *33*, 155–173. [CrossRef] [PubMed]
25. Tamura, Y.; Nishiyama, Y.; Wang, C.; Sun, Y.; Shurbevski, A.; Nagamochi, H.; Akutsu, T. Enumerating chemical graphs with mono-block 2-augmented tree structure from given upper and lower bounds on path frequencies. *arXiv* **2020**, arXiv:2004.06367.
26. Yamashita, K.; Masui, R.; Zhou, X.; Wang, C.; Shurbevski, A.; Nagamochi, H.; Akutsu, T. Enumerating chemical graphs with two disjoint cycles satisfying given path frequency specifications. *arXiv* **2020**, arXiv:2004.08381.