2018 IEEE Conference on Multimedia Information Processing and Retrieval

PharmKi: A Retrieval System of Chemical Structural Formula Based on Graph Similarity

Jingwei Qu $^{\#1}$, Xiaoqing Lu $^{\#*2}$, Chengcui Zhang $^{\dagger 3}$, Penghui Sun $^{\#4}$, Bei Wang $^{\#5}$, Zhi Tang $^{\#*6}$
# *Institute of Computer Science & Technology, Peking University, Beijing, P.R.China*
* *State Key Laboratory of Digital Publishing Technology, Beijing, P.R.China*
† *Department of Computer Sciences, University of Alabama at Birmingham, Birmingham, USA*
^{1,2,4,6} {*qujingwei, lvxiaoqing, sph, tangzhi*}*@pku.edu.cn*, ^{3} *czhang02@uab.edu*, ^{5} *wangbei@bupt.edu.cn*

***Abstract*—Different from conventional media type, chemical structural formula (CSF) is a primary search target as a unique identifier for each compound in the research field of medical information retrieval. This paper introduces a graph-based CSF retrieval system, PharmKi, accepting the photos taken from smartphones and the sketches drawn on tablet PCs as inputs. To establish a compact yet efficient hypergraph representation for molecules, we propose a graph-isomorphism-based algorithm for evaluating the spatial similarity among graphical CSFs, as well as selecting dominant acyclic subgraphs on the basis of overlapping analysis. The results of comparative study demonstrate that the proposed method outperforms the existing methods with regard to accuracy and efficiency.**

***Keywords*-chemical structural formula; multimedia information retrieval; frequent subgraph mining; graph isomorphism;**

I. Introduction

Medicine information retrieval is high-valued to health professionals, people with medical conditions and the society at large. Being able to search medicine information efficiently by the similarity of molecule structure is not only helpful for pharmaceutical innovations but also essential for intellectual property protection. Different from conventional media type, chemical structural formula (CSF) is the ideal and precise identifier of a chemical compound at the molecular level. However, the structure-similarity-based search in almost all existing retrieval systems is far from satisfactory.

The most formidable challenge in current retrieval approaches is the lack of a highly compact and efficient description of CSFs and the corresponding similarity measurements. Representing CSFs with graphs is a common option, which maps atoms to vertices and bonds to edges. However, such traditional graphs easily lead to a cost-prohibitive graph matching for large molecules. Moreover, there is yet another fundamental challenge, subgraph overlapping, that hinders these methods from establishing the correct compact representation of an original CSF graph.

The contributions of our work are threefold:

- We introduce a complete workflow of the CSF retrieval system (called PharmKi), including multiple input methods;
- we theoretically investigate a critical problem that directly affects the efficiency of graph matching and propose a graph-isomorphism-based algorithm in our solution;
- we empirically evaluate our system using the available public dataset.

The rest of this paper is organized as follows: Section II summaries the related work in this field. Section III introduces PharmKi, including the workflow, collapsing methods, hypergraph construction, and similarity measure. Section IV illustrates our experiments and evaluation results. Section V concludes this paper.

II. Related Work

Recent approaches for CSF retrieval can be roughly divided into three categories, sequence-based, fingerprint-based, and graph-based.

Many biological and chemical data are expressed as sequential strings. Chemical languages, for example, SMILES (Simplified Molecular Input Line Entry System) [1] can represent molecular structures with symbol strings.

Many current applications in compound comparison and virtual screening rely on fingerprint similarity [2]. Molecular fingerprints encode properties of molecules through bit string comparisons. Topological fingerprints [3] converted the paths of molecular features linearly up to a given number of connecting bonds.

Graphs provide a generic data structure widely used in cheminformatics and bioinformatics [4]. Graph-based approaches could be classified into four subcategories, graph descriptor, similarity-based graph mining, graph embedding, and graph kernel.

The descriptors range in complexity from one-dimensional statistics [5], to two-dimension topological indices [6], and to complex three-dimensional descriptions [7]. Graph mining aims to search similar compounds or to predict physical and biological properties of molecules. Conventional techniques include maximum common subgraph [8], frequent graph mining [9], and edit distance [10]. Graph embedding means representing graphs with vectors. To tackle graph mining/matching with machine

0-7695-6354-6/18/$31.00 ©2018 IEEE
DOI 10.1109/MIPR.2018.00016

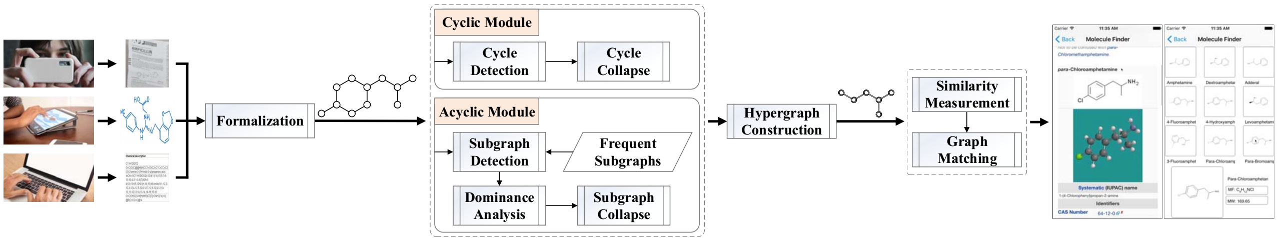


Figure 1. Workflow of CSF retrieval system, PharmKi: (i) CSF input and formalization, (ii) cyclic CSF collapsing, (iii) acyclic subgraph mining and optimization, (iv) construction of hypergraphs, and (v) similarity measurement.

learning methods [11], an explicit vector space is needed where each graph is embedded. Gibert et al. [12] adopted the attribute statistics and encoded the frequency of a node or an edge label. To avoid the severe drawback of graph embedding, i.e., loss of part of the structural information in the compact vector representation, graph kernels adopt a mathematical framework that defines a similarity measure between graphs as a scalar product in a Hilbert space [13], and therefore, provide an embedding space sufficiently large for graph transformation. Gaüzére et al. [14] adopted a bag of patterns defined as a subset of strict sub trees, which includes all labeled trees having at most six nodes. Cyclic pattern kernel [15] decomposed a graph into a cycle set and a set of bridges corresponding to atoms and bonds not in cycles. Learning more elaborate models, i.e. density and energy of compounds, improves the current analysis for realistic molecular systems effectively [16], [17].

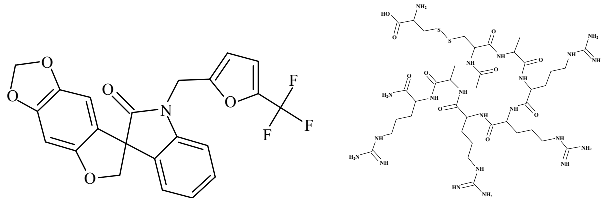
Overall, despite several decades of research on CSFs, there are still many challenges, including the cost-prohibitive matching of large molecules. In fact, most practical retrieval systems still rely on the indirect descriptions of the chemical structure. In academic research, some matching methods based on subgraph compression are proposed, but they are limited to several common substructures. Little attention is paid to the acyclic substructures in large-scale CSF datasets and retrieval.

III. Retrieval of CSFs

As shown in Fig. 1, the workflow of PharmKi consists of five key steps: (***i***) CSF input and formalization, (***ii***) cyclic CSF collapsing, (***iii***) acyclic subgraph mining and optimization, (***iv***) construction of hypergraphs, and (***v***) similarity measurement.

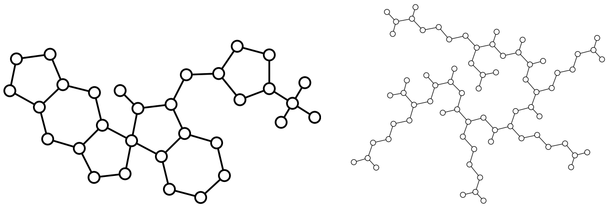
A. CSF Input and Formalization

In the first step, CSFs are obtained from multiple input channels. For a photo taken with a smartphone, we implement a rule-based method to extract the CSF in the picture, including image preprocessing, edge detection, character recognition, and formula construction. To acquire a CSF from a sketch drawn on touch screens, we developed several techniques specifically for this purpose, for instance, detecting corners based on temporal sequence analysis, distinguishing characters from lines by leveraging an interactive gesture, layout reconstruction via utilizing the domain knowledge.



(a) Funapide (b) Etelcalcetide

Figure 2. Examples of CSF: (a) a cyclic CSF; (b) an acyclic CSF.



(a) Graph of Funapide (b) Graph of Etelcalcetide

Figure 3. Graph representations of molecules (a) Funapide and (b) Etelcalcetide.

An obtained molecule is then represented by a labeled graph based on the following definition, where the vertices represent atoms and edges represent the bonds.

Definition III-A.1 A CSF is an undirected attributed simple graph represented as a 6-tuple $G = (V, E, \mathit{\Psi}_V, \mathit{\Psi}_E, \mathit{\Gamma}_V, \mathit{\Gamma}_E)$.

- V is a set of vertices, $E \subseteq V \times V$ is a set of edges;
- $\mathit{\Psi}_V$ and $\mathit{\Psi}_E$ are the label functions that assign labels to vertices in V and edges in E, respectively;
- $\mathit{\Gamma}_V$ and $\mathit{\Gamma}_E$ are the label sets of V and E, respectively.

Fig. 3 shows graph representations of molecules Funapide and Etelcalcetide (Fig. 2). To improve the efficiency of retrieval, we divide molecules into two categories, cyclic CSFs and acyclic CSFs (as shown in Fig. 2), by judging if they contain at least one cycle or not.

B. Cyclic Subgraph Collapsing

Given a cyclic CSF, denoted as a graph $\widehat{G} = (\widehat{V}, \widehat{E}, \Psi_{\widehat{V}}, \Psi_{\widehat{E}}, \Gamma_{\widehat{V}}, \Gamma_{\widehat{E}})$, the main steps of collapsing cycles in the cyclic CSF are:

1) Utilize a method based on the RDKit [18] to extract all the cycles in the cyclic CSF and form its cycle set, S_c;
2) Analyze the relations among the cycles in S_c, including separation, tangency, and intersection;
3) Establish a cycle graph G_c, in which each vertex corresponds to a cycle, and each edge represents the relation between every two cycles if they are spatially adjacent.

C. Acyclic Subgraph Mining and Optimization

Different from cyclic CSFs that share an apparent substructure, i.e., the cycle structure, most substructures in acyclic CSFs are full of diversity. More specific analyses, including the detection of frequent subgraphs, the selection of dominant subgraphs and acyclic graph collapsing, are worth exploring for acyclic subgraphs.

Definition III-C.1 A frequent subgraph is represented as a 6-tuple $G_f = (V_f, E_f, \Psi_{V_f}, \Psi_{E_f}, \Gamma_{V_f}, \Gamma_{E_f})$ with a specific *support* value $H(G_f)$.

- V_f is a set of vertices, $E_f \subseteq V_f \times V_f$ is a set of edges;
- $\varepsilon_{min} \leq |V_f| \leq \varepsilon_{max}$, ε_{min} and ε_{max} are the lower and upper bounds of the scale of G_f, respectively;
- $H(G_f)$ is the number of graphs (in a given graph dataset D) in which G_f is an induced subgraph, $H(G_f) \geq \delta$, i.e., δ is the lower bound of $H(G_f)$.

The obtained frequent subgraphs from $D = \{G_1, G_2, \ldots, G_i, \ldots, G_n\}$ are defined as another graph set,

$$S_f = \{{G_f}^1, {G_f}^2, \ldots, {G_f}^j, \ldots, {G_f}^m\} \tag{1}$$

where ${G_f}^j$ denotes the j^{th} frequent subgraph, $j \leq m$, and m is the total number of obtained frequent subgraphs. Λ_i is the feature vector to represent each graph $G_i \in D$:

$$\Lambda_i = [\lambda_i^{{G_f}^1}, \lambda_i^{{G_f}^2}, \ldots, \lambda_i^{{G_f}^j}, \ldots, \lambda_i^{{G_f}^m}] \tag{2}$$

where $\lambda_i^{{G_f}^j}$ is the number of ${G_f}^j$ instances in G_i.

Theoretically, we can collapse each frequent subgraph into a hypervertex to generate a hypergraph. However, the overlap between subgraphs poses a serious barrier to achieving the optimal hypergraph. Some pairs of frequent subgraphs share one or more vertices/edges: (*i*) *intersecting*, i.e., one subgraph partially overlaps with another; (*ii*) *including*, i.e., one subgraph is completely included by another. Furthermore, when $\lambda_i^{{G_f}^j} > 1$, multiple instances of the same frequent subgraph ${G_f}^j$ may overlap with each other.

Different selection strategies will lead to different subgraphs to be compressed and consequently different final hypergraphs. Without loss of generality, the minimum hypergraph among all possible results is considered optimal for further comparison as it leads to the lowest cost of matching calculation. Therefore, we transform this selection to a problem of optimal subgraph cover.

A dominance-priority algorithm based on the analysis of overlapping subgraphs is proposed as our solution to the cover problem. We define the *dominance* score ω for each frequent subgraph ${G_f}^j$ based on the scale $|{V_f}^j|$ and the *support* value $H({G_f}^j)$:

$$\omega({G_f}^j) = |{V_f}^j| + H({G_f}^j) \Big/ \sum_{j=1}^{m} H({G_f}^j) \tag{3}$$

$$H({G_f}^j) = \sum_{i=1}^{n} h({G_f}^j, G_i), \ h({G_f}^j, G_i) = \begin{cases} 1, \ \lambda_i^{{G_f}^j} \geq 1 \\ 0, \ Otherwise. \end{cases}$$

$$\Omega = \{\omega({G_f}^1), \omega({G_f}^2), \ldots, \omega({G_f}^j), \ldots, \omega({G_f}^m)\} \tag{4}$$

$\forall \omega({G_f}^j) \in \Omega$ reveals not only the local dominance with the scale of ${G_f}^j$, but also the global influence with $H({G_f}^j)$ over the entire graph dataset.

The dominant subgraphs in a graph are selected with Algorithm 1 in the following steps. We initialize all the vertices in a graph G_i as uncovered. In Steps 2∼3, a candidate frequent subgraph set ${S_f}^{can}$ is constructed by including all the frequent subgraphs in G_i, i.e., including only those ${G_f}^j$ whose corresponding $\lambda_i^{{G_f}^j} \neq 0$, then sorted in the decreasing order of their *dominance* scores in Ω. Then the dominant subgraphs set ${S_f}^{dom}$ is initialized (Step 4). To locate all possible instances of every frequent subgraph in G_i, we adopt a graph isomorphism algorithm, VF2 [19], to obtain multiple instances set $\Upsilon({G_f}^x, G_i)$ of each candidate frequent subgraph ${G_f}^x$ of ${S_f}^{can}$ in G_i in Step 6. During Steps 7∼16, each subgraph instance ${{G_f}^x}_y$ in the $\Upsilon({G_f}^x, G_i)$ of the current frequent subgraph ${G_f}^x$ is checked for its coverage (Step 8). If none of the vertices of ${{G_f}^x}_y$ are covered, label these vertices in G_i as covered, and insert the subgraph ${{G_f}^x}_y$ to ${S_f}^{dom}$ (if it is not already included) (Steps 9∼12). Otherwise, if at least one vertex of the current subgraph instance ${{G_f}^x}_y$ is already covered, we discard this instance and move on to the next instance in $\Upsilon({G_f}^x, G_i)$ (Step 14).

D. Construction of Hypergraph

The hypergraph G_h of a CSF takes into account the adjacency relations between cyclic and acyclic parts of the CSF. It can be defined as follows.

Definition III-D.1 An undirected attributed hypergraph is a 6-tuple $G_h = (V_h, E_h, \Psi_{V_h}, \Psi_{E_h}, \Gamma_{V_h}, \Gamma_{E_h})$ based on an undirected attributed simple graph $G = (V, E, \Psi_V, \Psi_E, \Gamma_V, \Gamma_E)$.

- V_h is a set of hypervertices. A hypervertex encodes at least one vertex $v \in V$. $E_h \subseteq V_h \times V_h$ is a set of hyperedges.

To establish the $\widehat{G}_h$ from a cyclic graph $\widehat{G}$ that has an associated cycle graph G_c, two auxiliary sets V_{ac} and E_{ac}

Algorithm 1 Dominant subgraph selection.

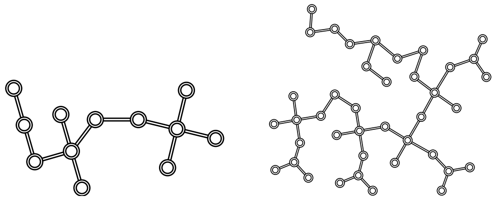
```
Input:  $G_i = (V_i, E_i, \mathit{\Psi}_{V_i}, \mathit{\Psi}_{E_i}, \mathit{\Gamma}_{V_i}, \mathit{\Gamma}_{E_i}) \in D$
        $S_f = \{{G_f}^1, {G_f}^2, \ldots, {G_f}^j, \ldots, {G_f}^m\}$
        $\mathit{\Lambda}_i = [\lambda_i^{{G_f}^1}, \lambda_i^{{G_f}^2}, \ldots, \lambda_i^{{G_f}^j}, \ldots, \lambda_i^{{G_f}^m}]$
        $\mathit{\Omega} = \{\omega({G_f}^1), \omega({G_f}^2), \ldots, \omega({G_f}^j), \ldots, \omega({G_f}^m)\}$.
Output: Dominant subgraphs in $G_i$.
Initialize all the vertices in $V_i$ as 'uncovered';
Filter: include all the ${G_f}^j$ of $S_f$ for which $\lambda_i^{{G_f}^j} \neq 0$ to obtain
    a candidate frequent subgraph set ${S_f}^{can}$;
Rank: Sort the subgraphs in ${S_f}^{can}$ in the decreasing order of
    their dominance scores in $\mathit{\Omega}$;
Initialize ${S_f}^{dom} = \emptyset$;
for each ${G_f}^x \in {S_f}^{can}$ do
    Compute $\mathit{\Upsilon}({G_f}^x, G_i)$ of ${G_f}^x$ in $G_i$ by VF2;
    for each ${G_f}^x{}_y \in \mathit{\Upsilon}({G_f}^x, G_i)$ do
        if all the vertices in ${V_f}^x{}_y$ are uncovered then
            Label ${V_f}^x{}_y$ in $G_i$ as covered;
            if ${G_f}^x{}_y \notin {S_f}^{dom}$ then
                ${S_f}^{dom}$.append(${G_f}^x{}_y$);
            end if
        else
            Discard ${G_f}^x{}_y$;
        end if
    end for
end for
```

are introduced, which include all the vertices and edges in $\widehat{V}$ and $\widehat{E}$ but not in any cycle, respectively. $\widehat{G}_h$ is initialized as G_c. Then $\widehat{V}_h$ is established by repeatedly adding vertices in V_{ac}. Thereafter, we determine whether each edge in E_{ac} connects to any cycle in G. If the edge connects to one cycle, we insert a corresponding hyperedge into $\widehat{E}_h$ with a label assigned by $\mathit{\Psi}_{\widehat{E}_h}$ to connect one hypervertex from V_{ac} and the other hypervertex that represents the corresponding cycle. All the other edges in E_{ac} that do not connect to any cycle will be inserted into $\widehat{E}_h$. Fig. 4(a) shows the hypergraph of molecule Funapide (Fig. 2(a)).

With the obtained dominant subgraphs in Algorithm 1, we generate the hypergraph $\widetilde{G}_h$ for an acyclic CSF $\widetilde{G}$ according to the following steps. We first collapse the covered subgraphs in ${S_f}^{dom}$ into the corresponding hypervertices in $\widetilde{V}_h$. Then the remaining uncovered vertices of $\widetilde{V}$ are added into $\widetilde{V}_h$. Next, we check whether each uncovered edge connects to any covered subgraph in $\widetilde{G}$. If the edge connects to one or two covered subgraphs, we insert a corresponding hyperedge with a label assigned by $\mathit{\Psi}_{\widetilde{E}_h}$ into $\widetilde{E}_h$ for two hypervertices of $\widetilde{V}_h$ that represent two adjacent subgraphs in the original $\widetilde{G}$. Finally, the remaining uncovered edges are added into $\widetilde{E}_h$. The hypergraph of molecule Etelcalcetide (Fig. 2(b)) is shown in Fig. 4(b).

E. Similarity Measure and Dual-stage Matching

We propose two measures of similarity between a query Q and each CSF G in a given dataset, η_e and η_l: $(\boldsymbol{i})$ η_e



(a) Hypergraph of Funapide

(b) Hypergraph of Etelcalcetide

Figure 4. Hypergraph representations of molecules (a) Funapide and (b) Etelcalcetide.

is the similarity on the element level, which describes the matching degree on atoms and bonds between Q and G:

$$\eta_e(Q, G) = MGED(Q, G) \tag{5}$$

where $MGED(Q, G)$ denotes the minimum graph edit distance between Q and G, calculated by a method based on a systematic strategy [20]. $(\boldsymbol{ii})$ A (sub)isomorphism degree η_l is the similarity on the layout level:

$$\eta_l(Q, G) = ISO(Q, G) \tag{6}$$

where $ISO(Q, G)$ denotes the number of (sub)isomorphism between the two graphs. We combine η_e and η_l by a weighted sum:

$$\eta(Q, G) = \alpha\eta_e(Q, G) + (1 - \alpha)\eta_l(Q, G) \tag{7}$$

where α and $1 - \alpha$ denote the weights for η_e and η_l, respectively.

A dual-stage matching is performed, including hypergraph-based screening and graph-based searching. In the first stage, query Q is compared with all the CSFs in the dataset based on hypergraph-based representation with η_e. Then the obtained top-k_1 ($k_1 \ll |D|$) results are utilized to build a candidate CSF set for the next stage. In the second stage, Q is compared with each CSF in the candidate set based on their original graph representations using η to obtain the top-k_2 CSFs as the final matching results.

IV. Experiment and Evaluation

To evaluate the performance of PharmKi, we first compare PharmKi with Wikipedia Chemical Structure Explorer (WCSE)[1] [21] on retrieval accuracy over Wikipedia molecules dataset (WIKI)[2]. Second, several retrieval cases are presented to intuitively compare with WCSE. In addition, we evaluate retrieval efficiency and collapsing efficiency on WIKI.

All experiments are conducted on an iMac with a 3.2GHz Intel Core i5 CPU and a 16 GB memory using MATLAB R2016b. We adopt the WIKI dataset for comparison. The number of molecules in this dataset is 15,312. The average, the maximum, and the minimum values of the number of

[1] http://www.cheminfo.org/wikipedia/

[2] http://www.cheminfo.org/wikipedia/smiles.txt

Table I
MAP, *DCG*, *RBP*85, AND *ERR* AT TOP-k_2 (2∼6) RESULTS

Metric	PharmKi	WCSE	Metric	PharmKi	WCSE
MAP_2	96.00%	91.50%	DCG_2	18.12	17.38
MAP_3	89.67%	84.33%	DCG_3	19.67	18.82
MAP_4	85.13%	79.88%	DCG_4	20.71	19.78
MAP_5	82.20%	76.90%	DCG_5	21.58	20.50
MAP_6	79.50%	73.25%	DCG_6	22.29	21.11
$RBP85_2$	26.73%	25.58%	ERR_2	94.60%	93.20%
$RBP85_3$	35.07%	33.17%	ERR_3	94.84%	93.49%
$RBP85_4$	41.66%	39.29%	ERR_4	94.94%	93.61%
$RBP85_5$	47.18%	44.38%	ERR_5	95.00%	93.67%
$RBP85_6$	51.57%	48.04%	ERR_6	95.03%	93.71%

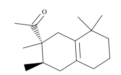
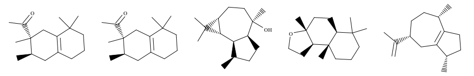
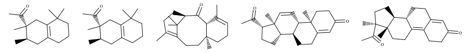
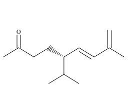
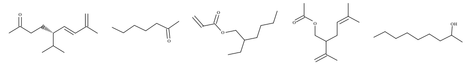
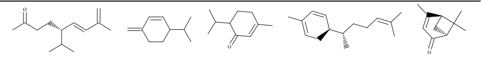
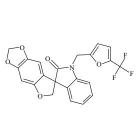
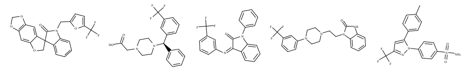
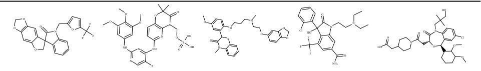
atoms in a molecule are 41.75, 2,794, and 1, respectively. The average, the maximum, and the minimum values of the number of bonds in a molecule are 42.83, 2,821, and 0, respectively.

Frequent subgraphs are extracted from the dataset, with the following parameter setting $\varepsilon_{min} = 3$ and $\varepsilon_{max} = 300$ (Definition III-C.1). To find an appropriate values of δ (subgraph frequency), we calculate the numbers of obtained frequent subgraphs at different frequency thresholds. We then set $\delta = 10\%$ in the WIKI dataset, because the number of frequent subgraphs starts to drop very slowly with the increase of δ passing 10%. After comparison of various parameter settings, we set $\alpha = 0.75$ (Equation 7) and $k_1 = 50$ for the first-stage matching (subsection III-E).

We compare PharmKi with WCSE [21], which is the state-of-the-art system allowing CSF similarity searching within WIKI dataset. As shown in Table I, Mean Average Precision (*MAP*), Discounted Cumulative Gain (*DCG*), Rank-Biased Precision (*RBP*), and Expected Reciprocal Rank (*ERR*) are adopted to evaluate the retrieval performance of the two systems. The parameter p in metric *RBP* is selected as 0.85. We set the range of values for k_2 to 2∼6. From the top-2 to the top-6 retrieved results, PharmKi achieves higher values on these metrics than WCSE at each one. Specifically, PharmKi not only retrieves more similar CSFs, but also obtains better ranking results than WCSE according to the results of *MAP* and *DCG*. Higher values on *RBP*85 and *ERR* of PharmKi indicate that it returns more desirable and satisfying CSFs than WCSE for users. As the top-k_2 increases, the performance deteriorates on all metrics. It indicates that both systems are able to assign a higher rank to more similar CSFs. Besides, the differences of *MAP*, *DCG*, and *RBP*85 of the two systems grow larger as top-k_2 increases, which demonstrates that PharmKi performs better than WCSE especially in large-scale retrieval of CSFs.

Several concrete retrieval results returned by PharmKi and WCSE are presented in Table II for intuitive comparison, with $k_2 = 5$. Both systems can find the exact match of the query CSF and return it as the top-1 result. However, in the remaining results: (*i*) The first query molecule, Iso E Super, contains two dominant cycles. PharmKi retrieves three similar CSFs with two cycles. However, WCSE only returns one CSF with two cycles, while the others are not very similar to the query. Besides, the atom number and the bond number of Iso E Super are 43 and 44, respectively. The average numbers of atoms and bonds among the PharmKi results are 42.4 and 43.8, respectively, but the corresponding mean values of WCSE are 47.8 and 49.8, respectively. This fact reflects that our results are closer to Iso E Super than those of WCSE. (*ii*) The second query molecule, Solanone, is an acyclic CSF, which is constituted by a dienone and two substitutes, a 5-propyl and a 8-methyl. The results of PharmKi are all acyclic CSFs including the carbon chains like Solanone. The results of WCSE, on the contrary, are all cyclic CSFs except the first one. (*iii*) The third query molecule, Funapide, contains six cycles and a trifluoromethyl. The four results of PharmKi all contain a trifluoromethyl, and they have three or four cycles. However, WCSE returns only one CSF with a trifluoromethyl. The average numbers of atoms and bonds of the top-5 results returned by PharmKi are 44.4 and 47.4, which are closer to the numbers of atoms and bonds in Funapide, 45 and 50, compared with the values of WCSE, 63.4 and 66.8, respectively. Unsatisfactory results of WCSE may be caused by the complicated structure of Funapide.

Table II
COMPARISON OF THE TOP-5 RETRIEVAL RESULTS FROM PHARMKI AND WCSE ON WIKI

Query	Top-5 retrieval results
	
	
	
	
	
	

For each query, the first row of column '**Top-5 retrieval results**' contains the top-5 results from PharmKi, and the second row shows the top-5 results of WCSE.

We further compare the retrieval time for queries of PharmKi with a baseline method. The baseline method means that a query is represented by a original graph, no collapsing or hyergraph representations. We randomly select 50 cyclic queries and 50 acyclic queries in WIKI. For each method, we measure the total retrieval time of the 100 queries with $k_2 = 10$ denoted as $T_{PharmKi}$ and $T_{Baseline}$, respectively. The results, $T_{PharmKi} = 15.33s$

and $T_{Baseline} = 90.06s$, show that PharmKi yields 82.98% less retrieval time. To evaluate the performance of the hypergraph, we compare the average values of $|V|$ and $|E|$ of all graphs and the same values of all corresponding hypergraphs in WIKI. The hypergraph saves 46.29% and 53.96% space for vertices and edges, respectively.

V. Conclusions

This paper proposes a graph-similarity-based retrieval approach for CSFs. To obtain satisfactory retrieval results, we propose an isomorphism-based algorithm for dominant subgraph selection based on the subgraph overlapping analysis. Experiments demonstrate the effectiveness of the proposed approach. However, due to the size of large molecules and their complexity, there are still many problems worth further exploration, including advanced hypergraphs for large formulas, stereo graphs for representing the 3D information and high-efficiency graph matching methods for evaluation.

Acknowledgment

This work is supported by the projects of National Natural Science Foundation of China (No. 61472014, No.61573028, and No. 61673029) and it is also a research achievement of Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

References

[1] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[2] I. Muegge and P. Mukherjee, "An overview of molecular fingerprint similarity search in virtual screening," *Expert opinion on drug discovery*, vol. 11, no. 2, pp. 137–148, 2016.

[3] C. James and D. Weininger, "Daylight fingerprints. daylight theory manual. daylight chemical information systems," *Inc., Irvine, CA*, 2011, http://www.daylight.com/dayhtml/doc/theory/index.html.

[4] S. Pan and X. Zhu, "Graph classification with imbalanced class distributions and noise," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 1586–1592.

[5] L. B. Kier, L. H. Hall *et al.*, *Molecular structure description*. Academic, 1999.

[6] J. Galvez, R. Garcia-Domenech, J. V. de Julian-Ortiz, and R. Soler, "Topological approach to drug design," *Journal of chemical information and computer sciences*, vol. 35, no. 2, pp. 272–284, 1995.

[7] A. H. A. El-Atta, M. I. Moussa, and A. E. Hassanien, "Predicting activity approach based on new atoms similarity kernel function," *Journal of Molecular Graphics and Modelling*, vol. 60, pp. 55–62, 2015.

[8] J. W. Raymond and P. Willett, "Maximum common subgraph isomorphism algorithms for the matching of chemical structures," *Journal of computer-aided molecular design*, vol. 16, no. 7, pp. 521–533, 2002.

[9] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis, "Frequent substructure-based approaches for classifying chemical compounds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 8, pp. 1036–1050, 2005.

[10] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," *Image and Vision computing*, vol. 27, no. 7, pp. 950–959, 2009.

[11] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. V. Lilienfeld, A. Tkatchenko, and K. R. Mller, "Assessment and validation of machine learning methods for predicting molecular atomization energies." *Journal of Chemical Theory & Computation*, vol. 9, no. 8, p. 3404, 2013.

[12] J. Gibert, E. Valveny, and H. Bunke, "Graph embedding in vector spaces by node attribute statistics," *Pattern Recognition*, vol. 45, no. 9, pp. 3072–3083, 2012.

[13] H. Huo and M. Rupp, "Unified representation for machine learning of molecules and crystals," 2017.

[14] B. Gaüzére, L. Brun, and D. Villemin, "Two new graphs kernels in chemoinformatics," *Pattern Recognition Letters*, vol. 33, no. 15, pp. 2038–2047, 2012.

[15] T. Horváth, T. Gärtner, and S. Wrobel, "Cyclic pattern kernels for predictive graph mining," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 158–167.

[16] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K. R. Mller, "Bypassing the kohn-sham equations with machine learning," *Nature Communications*, vol. 8, no. 1, p. 872, 2017.

[17] L. Li, T. E. Baker, S. R. White, and K. Burke, "Pure density functional for strong correlation and the thermodynamic limit from machine learning," *Phys.rev.b*, vol. 94, no. 24, 2016.

[18] G. Landrum, "Rdkit: Open-source cheminformatics," *Online. http://www.rdkit.org. Accessed*, vol. 3, no. 04, p. 2012, 2006.

[19] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub) graph isomorphism algorithm for matching large graphs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 10, pp. 1367–1372, 2004.

[20] W. Zheng, L. Zou, X. Lian, D. Wang, and D. Zhao, "Efficient graph similarity search over large graph databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 964–978, 2015.

[21] P. Ertl, L. Patiny, T. Sander, C. Rufener, and M. Zasso, "Wikipedia chemical structure explorer: substructure and similarity searching of molecules from wikipedia," *Journal of cheminformatics*, vol. 7, no. 1, p. 10, 2015.