Rev.Adv.Mater.Sci. 20(2009) 37-41

CHEMICAL SUBSTRUCTURE SEARCH SCREENING WITH FINGERPRINTS BUILT WITH SUBGRAPH ENUMERATION

Dmitry Pavlov and Igor Shturts

Applied Mathematics Department, St. Petersburg State Polytechnical University, Polytekhnicheskaya 29, 195251, St. Petersburg, Russia

Received: February 03, 2008

Abstract. The paper is aimed at efficient mass query optimization of substructure search on a large organic chemical database. Optimization method is based on so called fingerprints—compact bit arrays which represent graph structure in a packed form. Fingerprints allow cheap (but not complete) screening of fault cases, avoiding the subgraph isomorphism algorithm most of the time. Fingerprints, originally proposed by Daylight, are built in three independent sequential phases: (i) determining the characteristic features of a graph, (ii) hashing these features, and (iii) packing the hashes into a bit array. Our approach is novel in the first phase, in which we are using the edge subgraph enumeration, and in the second, in which we use the new graph hashing algorithm.

1. PROBLEM DEFINITION

Chemical structure of organic compounds is traditionally represented by a labeled graph, in which nodes are atoms, and edges are bonds between atoms, see e.g. Fig. 1.

Compound descriptions are stored in a relational database which can be considered as a simple array of molecules. Relational databases do not have any built-in functionality for chemistry or general graph-based data processing, so all the algorithms related to subgraph matching must be implemented in third-side software products, called cartridges. Well-known ones are MDL Direct and Daylight cartridges.

The substructure search problem is to find all the molecules in the database, which contain the given query molecule as a *substructure*. The most effective approach for mass query optimization is called *screening* and is quite simple: the most of false hits are screened before checking the exact subgraph match. Screening technique must be simple (otherwise, there is no point to do it instead of subgraph matching), and efficient (the more false hits are cheaply detected, the better).

The formal problem definition is the following. We denote by G the set of all graphs having labeled vertices and edges. Here is the definition of the labeled graph that we use:

$$G = (V, E, \alpha, \beta),$$

$$E \subset [V]^2 \to \alpha{:}V \to \Sigma_V \to \beta{:}E \to \Sigma_E,$$

$$\forall \{a,b\} \in E \quad a \neq b,$$

$$V \cap E = \varnothing \to \Sigma_V \cap \Sigma_E = \varnothing.$$

In our application area, Σ_V contains the element labels (C, N, H, O, S, Cl, Br, *etc.*), and Σ_E contains 4 chemical bond types (single, double, triple and aromatic). We say that graph $P = (V,E,\alpha,\beta)$ *contains* graph $P = (V',E',\alpha',\beta')$ and write $Q \to P$, if Q is isomorphic to some subgraph of P, i.e.:

Corresponding author: Dmitry Pavlov, e-mail: dmitry.pavlov@gmail.com

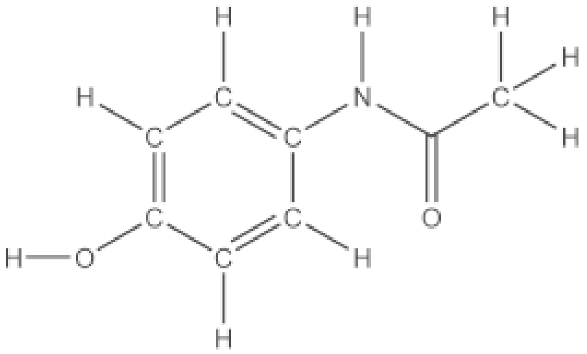

Fig. 1. Paracetamol structure contains the basic organic elements (C, N, O, H, *etc.*) and has single and double bonds.

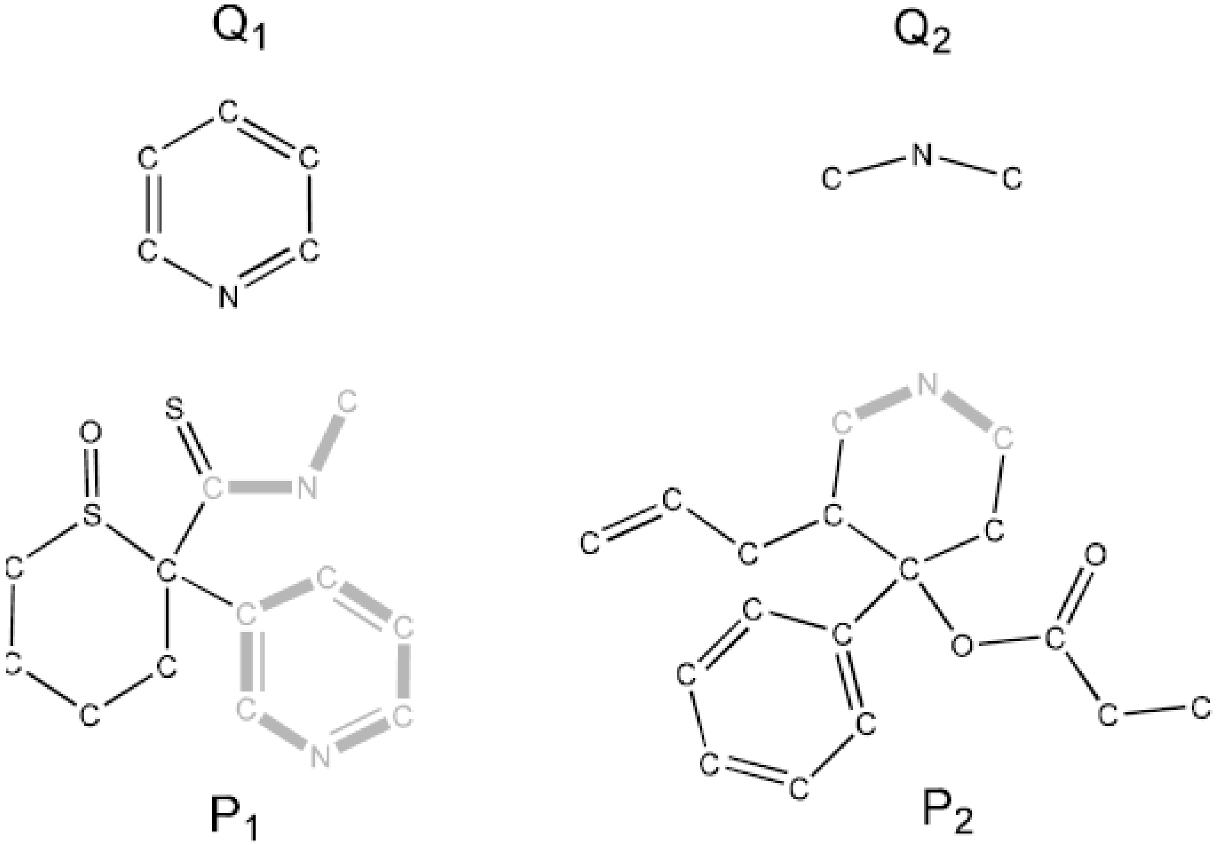

Fig. 2. Examples of subgraph matching.

$$
\begin{aligned}
&\exists \mu{:}V' \to V,\ \ \nu{:}E' \to E,\\
&\forall \nu_1,\nu_2 \in V',\ \ v_1 \neq \nu_2 \Rightarrow \mu(\nu_1) \neq \mu(\nu_2),\\
&\forall e_1,e_2 \in V',\ \ e_1 \neq e_2 \Rightarrow \nu(e_1) \neq \nu(e_2),\\
&\forall e = \{a,b\} \in E',\ \ \nu(e) = \{\mu(a),\mu(b)\} \in E,\\
&\forall \nu \in V',\ \ \alpha'(\nu) = \alpha(\mu(\nu)),\\
&\forall e \in E,\ \ \beta'(\nu) = \beta(\nu(e)).
\end{aligned}
$$

Given a query set $Q \subset G$ and the database $P \subset G$, we have to find matches $PQ = \{P \in P: Q \to P\}$ of each $Q \in Q$, see Fig. 2.

2. PREVIOUS WORKS

Since the subgraph isomorphism problem is NP-complete, and the result set of substructure search is often small in comparison to the database compound set, it is very important to screen as many as possible database compounds out before conducting the atom-by-atom subgraph matching. The first paper considering screening in substructure search was published in 1970 [1]. Since then, a typical size of chemical compound database grew up to millions, and a lot of research has been focused on the efficient screening algorithms.

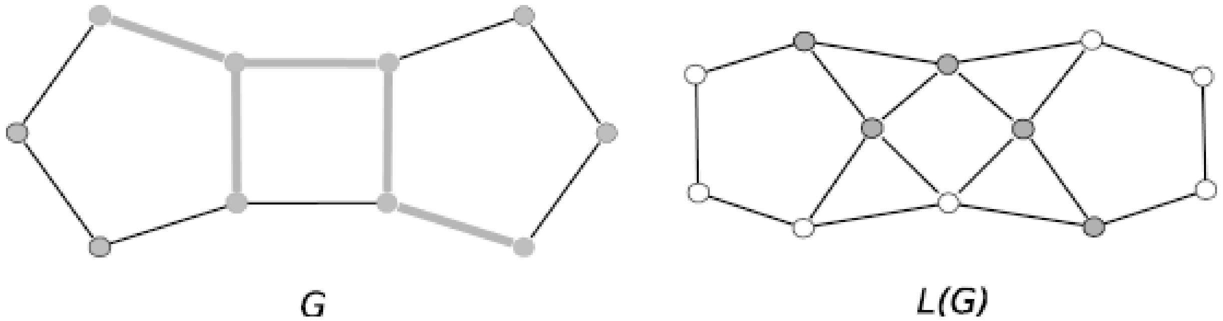

Fig. 3. Graph G and its line graph *L*(*G*). Highlighted vertices of *L*(*G*) are inducing the connected subgraph. The corresponding connected subgraph of *G* is also highlighted.

In 1979 MDL implemented the screening technique for their MACCS system. The algorithm was published in 2002 [2]. On the preprocessing stage the fixed-length bit vectors are calculated and saved for each compound in the database. When one uses the predefined set of descriptors that is common for all structures, the presence of each descriptor maps to one or more bits in the resulting bit vector. If the descriptor is absent, the corresponding bit(s) remain zero. Various structural keys may have some common bits to reduce the vector size. Screening is performed trivially by using “bitwise AND” operation on the structural key vectors of the query compound and the database compound.

In 1997, Daylight published the generalization of MDL structural keys, called *fingerprints* [3]. The main difference from the structural keys is that no predefined data is required for building fingerprints. Each bond chain in the graph of a chemical compound is considered as a descriptor. The bits that correspond to the descriptor are obtained with a pseudo-random number generator. The initial seed for this generator is calculated from the numerical hash of the string representation of the chain.

Fingerprints are more flexible than MDL structural keys; as they work equally well for all databases and all non-trivial queries, regardless of the exact chemical composition. The idea of the chain-based (or path-based) indexing is used in other subgraph searching algorithms, for example, APEX [4] and GraphGrep [5]. In addition to the screening algorithms based on precalculating some compact bit array for each compound in the database, there is another group of algorithms that are also worth mentioning. The general idea is building a tree-structured database index (in oppose to linear indexing by structural keys or fingerprints). The first paper proposing the tree-structured index was published in 1997 [6]. In the recent years, numerous papers have been devoted to the frequent subgraphs mining and to building the tree-structured indices from them [7-9].

The path-based indexing loses the structural information about cycles. The tree-based frequent structure indices do not lose structural information, but take much more resources to calculate, store, and manage large chemical databases. In the following section, we will propose the improvement for the Daylight technique which operates on subgraphs of limited size instead of chains of limited length.

3. ALGORITHM

The fingerprints building algorithm consists of three steps:
(i) determining the characteristic features of graph,
(ii) hashing these features, and
(iii) packing the hashes into a compact data structure.

The main drawback of Daylight fingerprints is that they lose a lot of structural information on stage (i). For example, the chains contained in a structure do not represent its rings. Fig. 3 shows that the information loss can occur even on very small and common chemical structures. The improvement that we are proposing is to use the *connected subgraphs* as characteristic features at stage (i) of fingerprint generation. This approach bears two algorithmic challenges. The first is the enumeration of the connected subgraphs, which is not as trivial as the chain enumeration. The second is the bit encoding of the subgraph. We will use the origi-

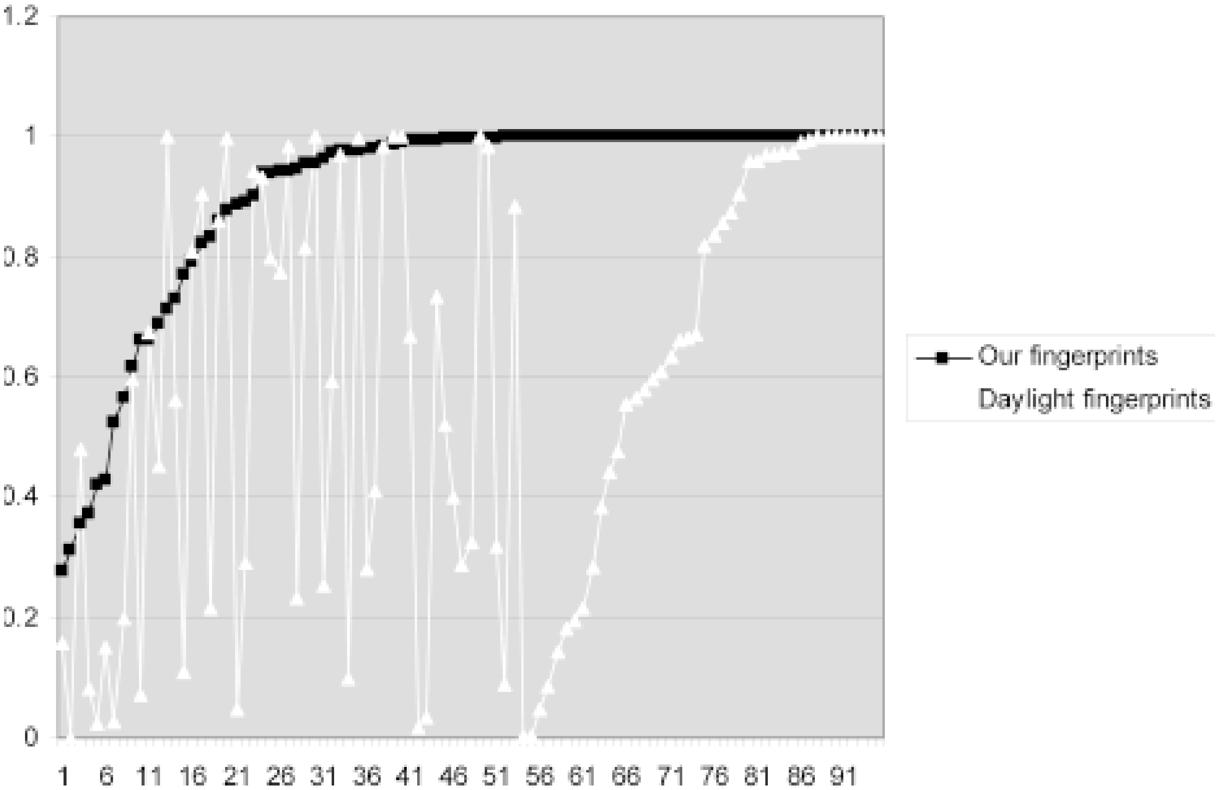

Fig. 4. Screening efficiency comparison. Points are sorted by screening efficiency of our algorithm.

nal Daylight's idea of pseudo-random bits, but the construction of the string for the initial seed should operate on graphs rather than on chains.

Our algorithm is based on the reverse search and related *induced subgraphs* enumeration algorithm proposed by Avis and Fukuda in 1992 [10]. The induced subgraph *G'* ∈ *G* is the subgraph defined (induced) with the subset of *G*'s vertices *V'* ⊂ *V* (*G*) All the edges of *G* having both ends in *V'* are present in *G'*. It is evident that an arbitrary connected subgraph of graph *G* is "induced" by the subset of *G*'s edges, which can be considered as vertices of line graph *L*(*G*) of *G*. Fig. 3 shows an example of the connected subgraph and the corresponding induced subgraph of the line graph.

Hence, the connected *G*'s subgraph enumeration procedure can be written similar to the induced subgraph enumeration, operating on *L*(*G*). Like the original reverse search algorithm, our modified version handles each subgraph once.

Encoding of the subgraphs

Each of the extracted subgraphs should be coded to a string. The numerical hash of this string should be given then as a seed to the pseudo-random number generator. There are two obvious requirements for the encoding scheme, first of which is obligatory, and the second one is desirable:

1. Isomorphic graphs must have identical codes.
2. Non-isomorphic graphs are desired to have different codes.

In terms of screening, the first requirement does not allow screening off true positive matches, and the second requirement educes the number of "false positives", yet it is theoretically impossible to eliminate them all. The encoding that meets both requirements is called the *canonical* one. The encoding that we propose is not canonical, i.e. it can give equal codes for some non-isomorphic graphs. Nonetheless, its discriminative ability is sufficient for chemical compound screening. It is also faster compared to the canonical encoding as it does not perform backtracking (practically unavoidable for canonical code computation).

Main algorithm

It calculates the fingerprint of a graph *G*, taking three numerical parameters: *K* – the size of the fingerprint, *L* – the subgraph size limit, and *p* – the number of bits being set for each subgraph. *K* = 6 makes hexagonal rings like that in Fig. 1 discriminative. Hexagonal and pentagonal rings are the most common kinds of rings in the chemical compounds, so setting *K* higher than 6 is inappropriate as seriously increases the amount of subgraphs and therefore slows down the fingerprints calculation. The *L* and p parameters do not affect the fingerprint calculaion performance, but they are essential for screening efficiency and database storage size. For best performance these parameters should be tuned for each database. For testing described in the next section, we chose *p* = 2 and *L* = 2560.

4. PRACTICAL RESULTS

The proposed algorithms were implemented within the Oracle cartridge designed for performing various types of search on stored chemical compounds, especially substructure search. Original Daylight fingerprints were also implemented as an option for performance comparison. Note that our implementation of Daylight fingerprints cannot be compared directly with performance of the Daylight cartridge.

The target set was the MDL ACD2D database, containing ~400,000 chemical compounds. The query set contained 95 structures used more-or-less frequently in the user retrieval. We compare two algorithms by screening efficiency—the quotient of hits number and the number of structures that passed the screening phase. The best possible screening efficiency ratio is 1, and it is achieved on some queries.

5. CONCLUSION

We have presented the solid improvement of the well-known Daylight screening technique. We have shown on practice that the structural information being lost by path-based fingerprints is dramatically important for the screening efficiency on the chemical databases. The further research may involve applying *variable-sized fingerprints* idea [3] to the improved fingerprint building algorithm.

REFERENCES

[1] J. E. Crowe, M. F. Lynch and W. G. Town // *J. Chem. Soc*. (c) (1970) 990.

[2] J. Durant, B. Leland, D. Henry and G. Nourse // *J. Chem. Inf. Comput. Sci.* **42** (2002) 1273.

[3] C. A. James, D. Weininger and J. Delaney, *Fingerprints – Screening and Similarity. Daylight Theory Manual* (Daylight Chemical Information Systems Inc., 1997), http://daylight.com/dayhtml/doc/theory/theory.finger.html.

[4] C.-W. Chung, J.-K. Min and K. Shim, *APEX: An adaptive path index for XML data*, In: *ACM SIGMOD International Conference on Management of Data* (2002), p. 121.

[5] R. Giugno and D. Shasha, *Graph Grep: a fast and universal method for querying graphs*, In: *International Conference on Pattern Recognition (ICPR)* (2002).

[6] K. Ozawa, T. Yasuda and S. Fujita // *J. Chem. Inf. Comput. Sci*. **37** (1997) 688.

[7] X. Yan and J. Han, *Span: graph-based substructure pattern mining*, In: International Conference on Data Mining (ICDM) (2002), p. 721.

[8] X. Yan, P. S. Yu and J. Han, *Graph indexing: a frequent structure-based approach*, In: ACM SIGMOD International Conference on Management of Data (2004). p. 134

[9] H. He and A.K. Singh, *Closure-tree: an index structure for graph queries*, In: *22nd International Conference on Data Engineering* (2006), p. 203.

[10] D. Avis and K. Fukuda // *Discrete Applied Math*. **6** (1996) 21.