An algorithm for graph isomorphism and its application to KEGG Compound Search

Nobuya Tanaka[1] **Susumu Goto** [1]
tanaka@kuicr.kyoto-u.ac.jp goto@kuicr.kyoto-u.ac.jp

Minoru Kanehisa[1]
kanehisa@kuicr.kyoto-u.ac.jp

[1] Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Kyoto 611–0011, Japan

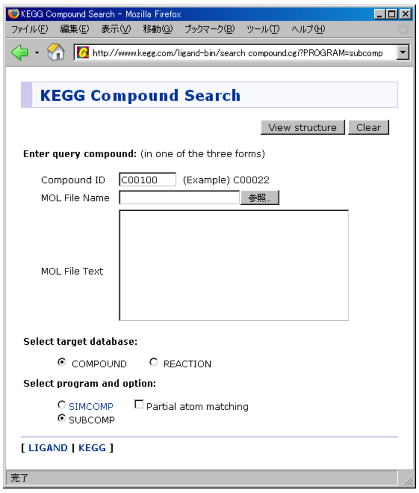
Keywords: subgraph isomorphism, pattern matching, compound search

1 Introduction

One of the significant challenges in bioinformatics is to represent and manipulate computationally complex biological information such as metabolic pathways, chemical compounds and protein-protein interactions. This challenge was successfully accomplished by representing biological information with graph objects[1].

On the other hand, these enormous graph structured databases (target graphs) require efficient algorithms for searching for a given subgraph (query graph). These kinds of search problems may be defined using the concept of subgraph isomorphism. Recently, subgraph isomorphism has come to be applied to biological analysis[2].

In this paper, we present a novel algorithm for solving the subgraph isormorphism problem by extending Ullmann's algorithm[3], which is currently recognized as the most efficient algorithm[4]. We implemented our algorithm as the SUBCOMP program which is available on the KEGG Compound Search page (Fig. 1).



http://www.kegg.com/ligand-bin/search compound.cgi?PROGRAM=subcomp

KEGG Compound Search

Enter query compound: (in one of the three forms)

Compound ID C00100 (Example) C00022

MOL File Name

MOL File Text

Select target database:

● COMPOUND ○ REACTION

Select program and option:

○ SIMCOMP □ Partial atom matching
● SUBCOMP

Figure 1: KEGG Compound Search

2 Method

Ullmann's algorithm tries to assign every possible node of the query subgraph to a node of the target graph by recursively generating a match matrix. In this algorithm, a technique called *refinement* is adopted in every recursion step in order to reduce the number of candidates. While Ullmann's algorithm tries to match all possible combinations of nodes, it often redundantly matches the same regions when there are permutable nodes in the query graph. These matches in effect cause redundant calculations that decrease the performance of the algorithm. Therefore, in order avoid such calculations, we developed a novel algorithm that dynamically prunes permutable nodes and reject undesired candidates in each recursion step.

In order to further optimize the *refinement* step, which is always most time consuming, the match and adjacency matrices were implemented with bit-vectors. A specialized bit-vector database for 32- and 64-bit computers was also developed.

3 Results and discussions

The KEGG Compound Database, which currently includes 11323 metabolic compounds, was transformed into a newly developed bit-vector database. The KEGG Compound Database may be searched using the SUBCOMP program and this database via the web[5] by selecting the SUBCOMP option (Fig. 1). Either a KEGG Compound ID or an MDL molfile may be used as a query graph.

In order to evaluate the efficiency of the SUBCOMP program, we randomly selected 70 compounds from the KEGG Compound Database with different numbers of atoms (1–70) as query compounds. The time required for searching the 70 compounds against the entire KEGG Compound Database was measured on a Celeron 1200MHz Gentoo Linux 2.4.2-Gentoo-r9 machine with 256MB of RAM (Fig. 2). The time for a single search did not exceed one second for the majority of the queries.

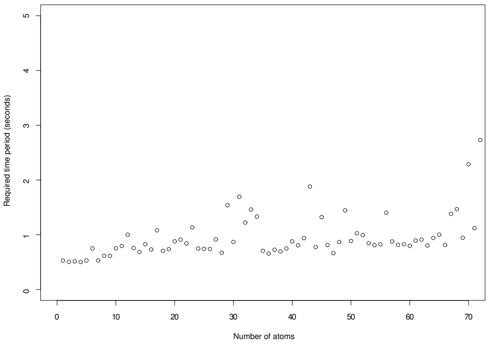


Figure 2: The time required for searching the 70 compounds against the entire KEGG Compound database.

4 Acknowledgements

This work was supported by grants from the Ministry of Education, Culture, Sports, Science and Technology, the Japan Society for the Promotion of Science, and the Japan Science and Technology Corporation. The computational resource was provided by the Bioinformatics Center, Institute for Chemical Research, Kyoto University.

References

[1] Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., and Hattori, M. The KEGG resources for deciphering the genome. *Nucleic Acids Res*, 32:D277–D280, 2004.

[2] Inokuchi, A., Washio, T., Okada, T. and Motoda, H. Applying the a priori-based graph mining method to mutagenesis data analysis. *J. Comput. Aided Chem.*, 2:87–92, 2001.

[3] Ullmann, J. R., An algorithm for subgraph isomorphism, *Journal of the Association for Computing Machinery*, 23:31–42, 1976.

[4] Barnard, J. M., Substructure searching methods: Old and new, *Journal of Chemical Information and Computer Science*, 33:532–538, 1993.

[5] `http://www.genome.jp/ligand-bin/search_compound.cgi?PROGRAM=subcomp`