Heuristics for Chemical Compound Matching

Masahiro Hattori hattori@kuicr.kyoto-u.ac.jp
Yasushi Okuno okuno@kuicr.kyoto-u.ac.jp
Susumu Goto goto@kuicr.kyoto-u.ac.jp
Minoru Kanehisa kanehisa@kuicr.kyoto-u.ac.jp
Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Kyoto 611-0011, Japan

Abstract

We have developed an efficient algorithm for comparing two chemical compounds, where the chemical structure is treated as a 2D graph consisting of atoms as vertices and covalent bonds as edges. Based on the concept of functional groups in chemistry, 68 atom types (vertex types) are defined for carbon, nitrogen, oxygen, and other atomic species with different environments, which has enabled detection of biochemically meaningful features. Maximal common subgraphs of two graphs can be found by searching for maximal cliques in the association graph, and we have introduced heuristics to accelerate the clique finding. Our heuristic procedure is controlled by some adjustable parameters. Here we applied our procedure to the latest KEGG/LIGAND database with different sets of parameters, and demonstrated the correlation of parameters in our algorithm with the distribution of similarity scores and/or the execution time. Finally, we showed the effectiveness of our heuristics for compound pairs along metabolic pathways.

Keywords: chemical compound, atom-typing, clique finding, similarity score

1 Introduction

In this post-genomic era, one of the most exciting goals is to understand systematic aspects of biology, that is, to elucidate how cellular functions result from intricate networks of molecular interactions [14]. Those functions are believed to be realized as a complex system rather than simply as a collected body of every molecular function. Coupled with this interest, significant efforts are undertaken for developing high-throughput experimental technologies and producing large-scale data in transcriptome, proteome, and metabolome analyses. Those genome scale data and up-to-date computational methods may give us the whole view of biological snapshots and lead us to try to uncover the complicated nature of biological events. Now, the sequence based methods for comparing genes or proteins are well established and we already have a limited view of the "gene universe" in terms of the number of ortholog groups as reported in COG [25]. Similarly, well-known methods for three-dimensional structure comparisons provide a landscape of the "protein universe" in terms of the number of unique folds in SCOP [19] or CATH [20]. On the contrary, we have little knowledge on the "chemical universe" consisting of chemical compounds and reactions in biological processes. In fact, biochemical characteristics of chemical compounds have not been studied well in a genome scale, and the classification of all metabolites has hardly been performed systematically so far, despite the fact that chemical compounds are as important as other macromolecules of proteins and nucleic acids in understanding molecular interaction systems.

The difference of the situation between a protein and a metabolite may arise from the absence of the effective similarity measure of chemical compounds. Thus, to achieve that end, it seems necessary to develop the method to compare chemical compounds accurately and rapidly. Of course, we know there have been several similarity measures between two chemical compounds [18, 29]. However, most

of them were inadequate to our needs. For instance, the comparison of bit strings has been used as the most famous method [5]. In this methodology, the information about a compound structure is reduced into a concatenation of several hundreds of bits [1]. A numerical vector method [3, 4] or fingerprint method [11, 27] have also been used in some applications, but they are just the mathematical extension of the bit-string comparison method. In contrast, the comparing two graphs of compounds directly by using the graph theory has become one of major categories of cheminformatics [21, 22, 23], but it has been still limited to the applications of small size problems because of its time-consuming nature. The graph comparison methods have the fundamental difficulty that the graph isomorphism problem is NP-hard and the computational time will increase exponentially.

In a recent work [9], we have developed a suite of new computational tools, to annotate an atomic environmental property for each atom of a biochemical compound, to rapidly identify common substructures between two compounds with a graph comparison method, and to evaluate statistical significance of similar substructures in a large dataset. Using these tools, we could include physicochemical information into the representation of atoms, which is commonly used to evaluate chemical aspects of compounds [6, 16, 17]. In addition, we introduced several heuristics into the algorithm of similarity calculations. Then, we could decrease the exponential difficulties of graph comparison methods to the practical level that could be tolerated, while holding high accuracies for graph similarities found. In this paper, we demonstrated the performance of our algorithm by applying it to comparison of about 10,000 compounds in the latest KEGG/LIGAND [7, 8] with different sets of changeable parameters. The results indicate how each parameter is correlated with the total distribution of similarity scores and/or other important factor, such as an execution time or the total accuracy of solutions. In particular, since we suppose this sort of methodology is much important for analyses of metabolic pathways, we should evaluate the effectiveness of our algorithm to detect closely related compound pairs containing large overlapping regions. Then, we applied our method to two different sets of chemical compound pairs, one is originated form the PATHWAY database in KEGG [12, 13], the other is made up at random from the whole set of chemical compounds. Thus, we could show the difference and the ability of our algorithm.

2 Materials

We have used chemical compound data in the COMPOUND section of the KEGG/LIGAND database (version 26.0 + update 2003/06/13). The total number of compounds with chemical structures is 10,001, roughly classified into 977 drug-related compounds, 2,649 phytochemical compounds (secondary metabolites in plants), and 6,375 metabolites and other compounds originating from the KEGG metabolic pathways or the enzyme nomenclature (EC number classification). We consider each chemical structure as a labeled 2D graph with atoms as its vertices and covalent bonds as its edges, excluding hydrogen atoms. We do not consider any 3D features and do not discriminate chirality. Some KEGG compounds are described in a generic form or a polymeric form, such as primary alcohol (R-OH) or starch ({C12H20O11}n), which is often necessary to make users easy to understand metabolic pathways. We treat these obscure compounds by the following rules: (i) the R group is just taken as 'R' atom, that is, as if R were the 69th atom type (in addition to the 68 types described in the Method section), and (ii) the degree of polymerization is taken as 1, which means any polymeric structures degenerate to corresponding monomers.

3 Method

3.1 Heuristics of Atom-Typing

The same atom species in chemical compounds must be discriminated by different labels, because they show different physicochemical properties according to their spatial and chemical circumstances. Such

labeling system is usually called as an atom-typing in cheminformatics. Here, we also introduce one of atom-typing as the vertex labeling function $p(v)$ into the graph representation of chemical compounds. This labeling function should reflect the environmental features of atoms, and is defined by a series of the examinations of: (i) whether the atom is a member of a ring structure or not, (ii) what types of bonds are connected to the atom; for example, single, double, triple and aromatic bonds, and (iii) which atoms are adjacent to the atom.

Our labeling system is just straightforward and can be generated computationally based only on the bond patterns of atoms and the functional groups that they belong to. Hence, each atom of all chemical compounds in KEGG could automatically be assigned as one of new labels from their initial graphs stored in the MDL/MOL file format. Fig. 1 shows the list of new atomic labels and corresponding environments. The atom types are usually coded in three letters, the first letter is the same with the original atom species, the second stands for the physical sense of orbits around the atom, and the last letter is assigned serially. In Fig. 1, carbon (C) is shown in diagram **a**, nitrogen (N) in **b**, oxygen (O) in **c**, sulfur (S) in **d**, phosphorus (P) in **e**, and the rest in **f**. In each diagram, H is a hydrogen atom and R indicates an atomic group larger than a simple H. The last category **f** is miscellaneous containing any C, N, O or S with no suitable class in **a**, **b**, **c** or **d**. A halogen is labeled as X, and other atoms are reduced into Z. Thus, the total number of atom types is 68.

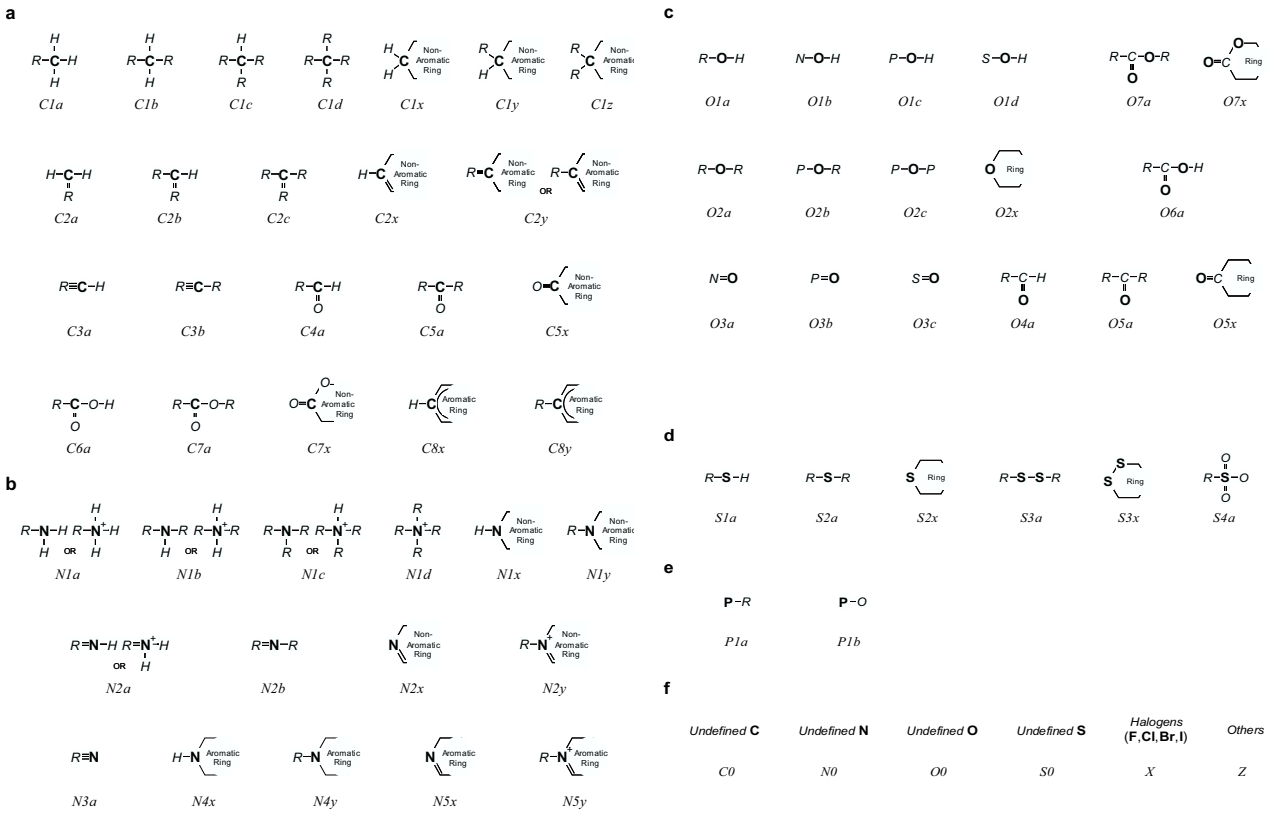
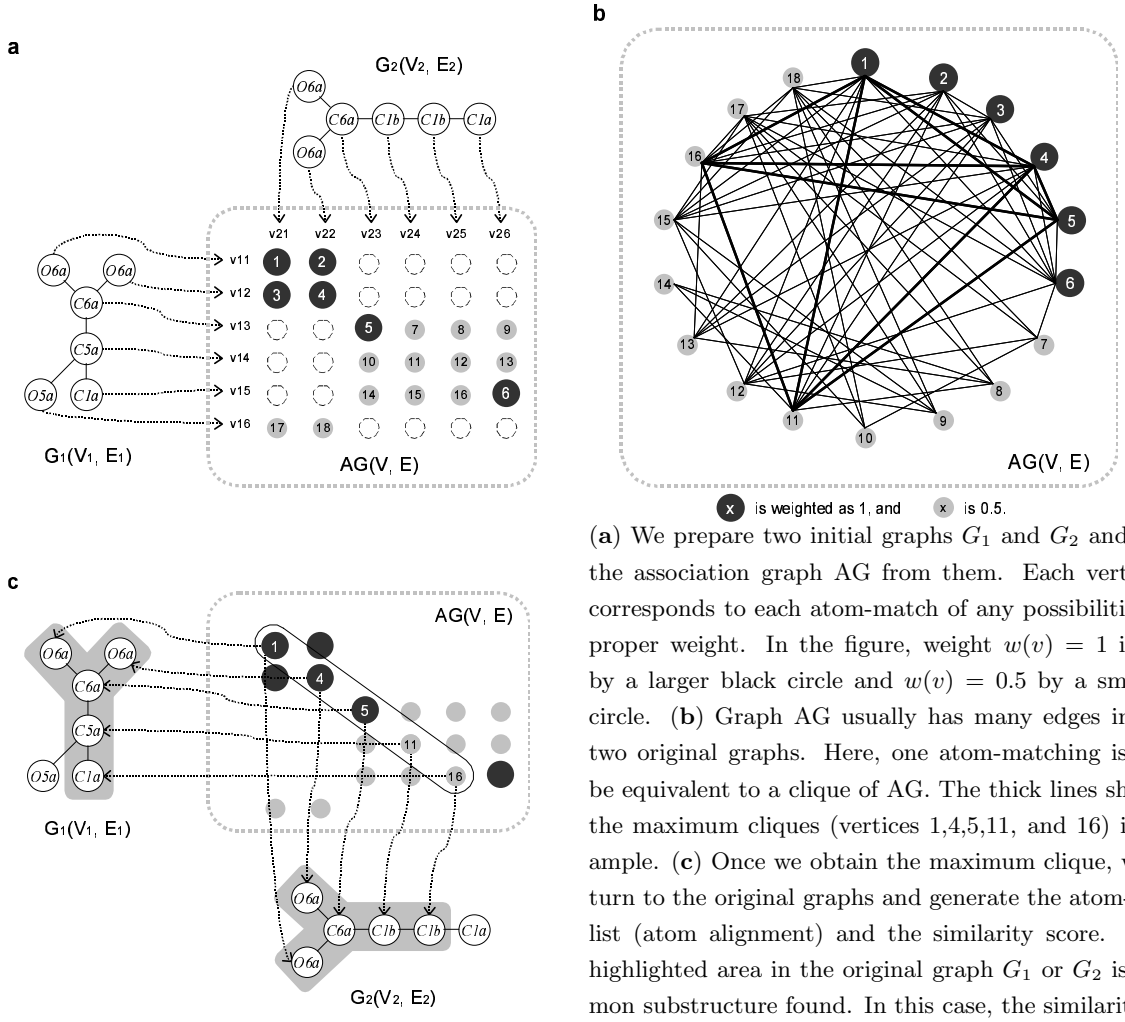


Figure 1: The list of 68 different atom types in our atom-typing.

3.2 Basic Algorithm for Comparing Compounds

Our approach to finding common (isomorphic) subgraphs is essentially the same as the traditional association graph method [15, 24]. Here we summarize the terminology for relevant graph features and the basic idea of this algorithm. Fig. 2 is a computational example when matching two chemical compounds in this algorithm.

(1) Maximum clique (MCL): A vertex-labeled graph consisting of the set of vertices V and the set



(**a**) We prepare two initial graphs G_1 and G_2 and calculate the association graph AG from them. Each vertex of AG corresponds to each atom-match of any possibilities, with a proper weight. In the figure, weight $w(v) = 1$ is denoted by a larger black circle and $w(v) = 0.5$ by a smaller gray circle. (**b**) Graph AG usually has many edges induced by two original graphs. Here, one atom-matching is found to be equivalent to a clique of AG. The thick lines show one of the maximum cliques (vertices 1,4,5,11, and 16) in this example. (**c**) Once we obtain the maximum clique, we can return to the original graphs and generate the atom-matching list (atom alignment) and the similarity score. Here, the highlighted area in the original graph G_1 or G_2 is the common substructure found. In this case, the similarity score is $5/(6+6-5) = 5/7 = 0.71$.

Figure 2: The flow of the algorithm.

of edges E is denoted by $G(V, E)$. A clique of graph G is defined as a complete subgraph in G. The maximum clique in graph G is the clique of G whose cardinality is not smaller than that of any other clique in G. The maximum clique of graph G is denoted as $\mathrm{MCL}(G)$.

(2) Maximal common subgraph (MCS) and simply connected common subgraph (SCCS): A subgraph of graph G is a new graph obtained from G by deleting some edges and vertices. A common subgraph of G_1 and G_2, $\mathrm{CS}(G_1, G_2)$, is a graph which is isomorphic to a subgraph of both G_1 and G_2. The maximal common subgraph of G_1 and G_2, $\mathrm{MCS}(G_1, G_2)$, is the $\mathrm{CS}(G_1, G_2)$ whose cardinality is not smaller than that of any other $\mathrm{CS}(G_1, G_2)$. A simply connected common subgraph, $\mathrm{SCCS}(G_1, G_2)$, is a $\mathrm{CS}(G_1, G_2)$ within which each vertex is connected to at least one other vertex. The $\mathrm{MCS}(G_1, G_2)$ must be a set of $\mathrm{SCCS}(G_1, G_2)$'s.

(3) Association graph (AG): The graph product $\mathrm{GP}(V, E)$ of two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ is a new graph defined on the vertex set $V = V_1 \otimes V_2$ and the set of edges $E = V \otimes V$. The association graph $\mathrm{AG}(V, E)$ defined here is one of the graph products with the following adjacency conditions. Any $e(v_{ij}, v_{st}) \in E$ is considered to be adjacent: (i) if $v_{1i} \in V_1$ is adjacent to $v_{1j} \in V_1$ in the original graph G_1 and $v_{2s} \in V_2$ is adjacent to $v_{2t} \in V_2$ in the original graphs G_2, or (ii) if v_{1i} is not adjacent to v_{1j} and v_{2s} is not adjacent to v_{2t}. Fig. 2b is a sample of AG derived from two chemical graphs G_1 and G_2 in Fig. 2a.

The association graph AG made by the previous definition should have all possibilities of vertex matches between two initial graphs G_1 and G_2; namely, a clique in AG corresponds to a common

subgraph between G_1 and G_2. Thus, the original problem of obtaining the largest match of atoms can be reduced to the computational task of searching for the largest clique in AG, MCL(AG). Here, the largest clique is defined as the largest sum of weights (see below). After calculating the largest clique with the largest weights, we can easily obtain the list of matching atoms, shown in Fig. 2c.

3.3 Heuristics of Weighting Atom Type Matches

In a conventional method, each vertex of the association graph is weighted as only one or zero, called all-or-none weighting here, depending on whether two vertices from the original graphs do or do not match. However, this type of weighting scheme is too strict for our representation where 68 atom types obviously share one of the 7 categories of atomic species. Thus, we took another system of weights formulated as follows. Given two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, the vertex v_{ij} of the association graph $\mathrm{AG}(V, E)$ is induced from two vertices $v_{1i} \in V_1$ and $v_{2j} \in V_2$ and is weighted as:

$$w(v_{ij}) = \begin{cases} 1, & if p(v_{1j}) = p(v_{2j}), \\ 0.5, & if p(v_{1j}) \neq p(v_{2j}) \quad and \quad atom(v_{1j}) = atom(v_{2j}), \\ 0, & otherwise \end{cases}$$

Of these three, the second statement is newly introduced here. We call this weighting scheme the loosely weighting through this paper. Even with this complex weighting scheme, we can still define the maximal common subgraph $\mathrm{MCS}(G_1, G_2)$ as the maximal clique MCL(AG) with the maximal sum of weights.

3.4 Heuristics in the Clique Finding Algorithm

The clique finding of a given graph is a well-studied problem and our implementation of the clique finding is a modified version of the Bron-Kerbosch algorithm [2]. Since the association graph $\mathrm{AG}(V, E)$ is generated only for the matching vertices in the initial graphs, the number of vertices in AG is much larger under the loosely weighted condition than the all-or-none condition. The difference of the number of vertices in AG is directly shown in Fig. 2 as large or small circles. The large black circles denote the vertices of AG originated from the all-or-none rule, and the small gray circles have been yielded through the loosely weighting statement ($w(v) = 0.5$). The larger number of vertices in AG results that the calculation based on this algorithm does not finish within a practical time for many compound pairs in our database. Thus, we need to incorporate further heuristics into the calculation.

First, we simply stop the calculation of clique finding after a reasonable number, R_{max}, of recursion steps in a recursive implementation of the B-K algorithm and obtain a candidate set of MCLs (maximal cliques), that is, MCSs (maximal common subgraphs). Then we start to search better common subgraphs, called quasi-MCSs, from the candidate set. In this second optimization step, we eliminate small SCCSs (simply connected common subgraphs) whose cardinality is smaller than a given threshold, S_{min}, and extend only other larger SCCSs. The SCCSs with small cardinality are frequently found as noises around the conserved structure of two compounds, such as separate matches of single atoms. Mathematically those separate matches should be considered to obtain the MCS, but the quasi-MCS without considering them may be biochemically meaningful. After the elimination of those small SCCSs, we search for and extend the other SCCSs one by one greedily until no more atom pairs can be included. Finally we obtain the quasi-$\mathrm{MCS}(G_1, G_2)$. In this heuristics, both R_{max} and S_{min} are controllable cutoff parameters.

3.5 Normalized Similarity Score

The maximal common subgraph $\mathrm{MCS}(G_1, G_2)$ is obtained by maximizing the number of matched atom types, which is a raw score that depends on the sizes of the original graphs G_1 and G_2. We introduce a normalized score, utilizing one of the most popular measures, the Jaccard coefficient [10, 26], also

known as the Tanimoto coefficient [24, 28, 30]. It is the ratio of the size of the common substructure (AND graph) divided by the size of the non-redundant set of all substructures (OR graph). The OR graph can be described as $G_1 + G_2 - MCS(G_1, G_2)$. Thus, the Jaccard coefficient JC(G_1, G_2) is formulated as:

$$JC(G_1, G_2) = \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|} = \frac{|MCS(G_1, G_2)|}{|G_1 + G_2 - MCS(G_1, G_2)|} = \frac{|MCS(G_1, G_2)|}{|G_1| + |G_2| - |MCS(G_1, G_2)|}$$

In the current implementation, we just obtain a quasi-MCS rather than an exact-MCS. Then, the above definition of JC will be rewritten from MCS to quasi-MCS.

4 Result and Discussion

4.1 Atom-typing and Loose Weighting

We have introduced a new atom labeling system into the representation of vertices in the original graphs and the loosely weighting system into the weighting of vertices in the association graph. Here, our atom-typing is based only on the bond pattern and physicochemical properties of atoms and their neighboring atoms. This means that we can calculate conversion from an atomic species to an atom type by a computational method without any supervisor knowledge, and that we can incorporate some of the three-dimensional information into atom types. Here, it is obvious that the complexity of atom-typing must be correlated with the complexity of matching vertices in graph comparisons. When the atom-typing becomes more complicated, the resulting graph contains a larger number of vertex types, and the common subgraph we obtain will become smaller if vertex matches are distinguished in an all-or-none manner. Thus, we assign an intermediate weight for those vertex mismatches consisting of the same atomic species, where the weighting is represented by an adjustable parameter. As the parameter value goes to 0, the computational result approximates to that of the all-or-none weighting rule. When it turns to 1, it will become the same as the result without any complicated vertex labels, that is, the result with original graphs without atom-typing. Hence, we can describe the general idea of this parameter as that playing a role in decreasing the complexity of atom-typing.

From these observations, the frequency of atom matching between two graphs is viewed as controlled by both the complexity of the atom-typing and the value of the intermediate weight. The sizes of common subgraphs found and the similarity scores are correlated with both of them. Here we changed only the weighting parameter to find an optimal value for the current scheme of our atom-typing. Figure 3 shows the comparison of three parameter values $w(v)$ =0.0, 0.5, and 1.0. Here $w(v) = 0.0$ corresponds to all-or-none weighting for the 68 atom types, and $w(v) = 0.0$ corresponds to all-or-none weighting for the 7 atom species. It is apparent that $w(v) = 0.0$ is not suitable because it detects only small regions consisting of perfect matches of 68 atom types. The performance of $w(v) = 1.0$, which is equivalent to normal graph comparison without atom-typing, cannot be estimated from this figure alone, but it probably contains many irrelevant matches from a biochemical point of view. In contrast, we expect that an intermediate weighting would detect optimal matches allowing mismatched atom types to be incorporated and removing superfluous matches of atom species. The performance of the intermediate weighting is indicated by the solid line where we show the case of $w(v) = 0.5$.

4.2 Heuristics in Clique Findings - R_{max}

The R_{max} parameter is one of the two cutoff parameters in our clique finding procedure, and influences the resulting solution in terms of how far it is from the exact solution and how long it takes to compute. If the R_{max}parameter is set to 1, which of course is not really meaningful, then the resulting solution will be entirely unsatisfactory, but the solution can be obtained instantaneously. When the R_{max}

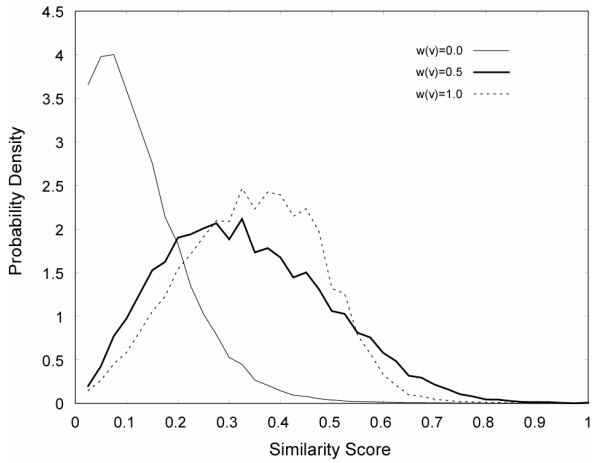


Figure 3: Distribution of scores with different weights.

parameter is set to infinity, then the resulting solution will be the exact solution, but the computation time may require an infinite amount of time. Here, we have examined the effect of the R_{max} parameter by changing the value from 10 to 10,000,000, and estimated the accuracy of solutions by the ratio of the matched sizes in heuristic and exact solutions. Of course, we could not tell if exact solutions (R_{max} = infinite) were really found, thus, we considered the results with R_{max} = 10,000,000 were identical or very close to the exact solutions. The result is shown in Fig. 4, where the closeness to the exact solution represented by the ratio of two solution sizes and the computation time required are plotted against the changing values of R_{max}.

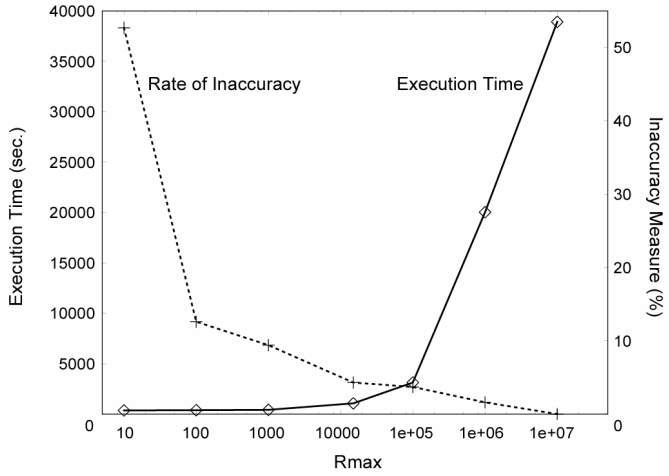


Figure 4: The effect of R_{max} on execution time and accuracy of solutions.

In this figure, the result with each R_{max} was obtained by comparing thousand pairs of chemical compounds, which had been randomly selected from the KEGG/LIGAND database. Thus, the execution time itself is not equivalent to the time required to compute all possible pairs in the database, like Fig. 3 or Fig. 5. We can observe the exponential tendency of computation time to find cliques, and the computation becomes impractical when R_{max} exceeds 100,000. In contrast, the inaccuracy of solutions exhibits an exponential behavior in an opposite way with decreasing R_{max}, where the inaccuracy is measured by the number of pairs among 1,000 pairs that contained solution sizes smaller than 90% of exact solutions (we also used 70%, 80%, and 100% as alternative measures). Since we assume that solutions with R_{max}= 10,000,000 are exact, the rate of failed solutions goes to 0 at R_{max} = 10,000,000. We conclude from Fig. 4 that an optimal value of R_{max} is between 10,000 and 100,000. In practice

we use $R_{max} = 15{,}000$.

4.3 Heuristics in Clique Findings - S_{min}

The S_{min} parameter is the other parameter in our clique finding algorithm. The original idea of introducing S_{min} comes from the observations that we had in preliminary applications of our method. We found that two or more larger SCCSs were sometimes separated by other small SCCSs, whose cardinality are 1 or a few. However, such larger SCCSs may better be connected as a single SCCS from a biochemical point of view. Therefore, we have introduced the S_{min} parameter into our procedure to favor connected components. When S_{min} is set to 1, any effect of this heuristics is not incorporated into the calculation. Thus, the S_{min} parameter is chosen to vary from 2 to MAX, where MAX is the size of the largest SCCSs. Fig. 5 shows the computational result with three different values of S_{min}: 1, 2 and MAX.

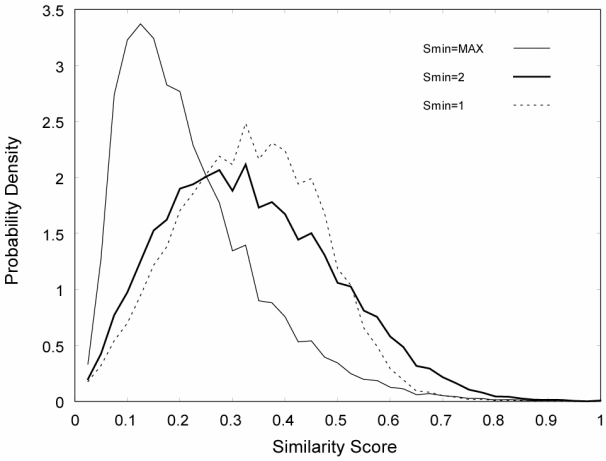


Figure 5: Distribution of scores with different S_{min}.

The S_{min} parameter controls the size of matches found with a given value of R_{max} as a result of eliminating small SCCSs and extending large SCCSs. The distributions of scores for all possible pairs are shifted to the left side when as S_{min} becomes larger, which is due to the elimination of small SCCSs. It is obvious that retaining only the largest SCCSs ($S_{min} = MAX$) is not appropriate, for it fails to detect too many matches. In contrast, considering all SCCSs ($S_{min} = 1$) is not practical either, for solutions obtained are apparently far from the exact solution in terms of the ratio of matched sizes. The fact that the tail of the distribution for $S_{min} = 2$ is higher than the tail for $S_{min} = 1$ indicates that solutions closer to the exact solutions are better found with the elimination and extension heuristics, that is, the elimination step makes the total distribution lowered and the extension step simultaneously enhances the right tail of the distribution.

4.4 Implications on Pathway Analysis

Since we have intended to apply this method to analyze metabolic pathways [9], we need to evaluate the effectiveness of our algorithm, in particular, for detecting biochemically related chemical compound pairs rather than randomly matched pairs. Then, we tested our method with two different sets of chemical compound pairs. One is originated from the KEGG/PATHWAY database in KEGG, and the other is made up at random from the whole set of chemical compounds stored in KEGG/LIGAND. Each dataset contains 1,000 compound pairs and the performance of our method is measured by the accuracy of solutions. Here, the accuracy is assessed by the similar way as in Fig. 4, and is defined by the ratio of solution size with $R_{max} = 15,000$ to that with $R_{max} = 10,000,000$. The result is shown in Table 1.

Table 1: Performance of our method with different datasets

Accuracy (%)	Pathway (pairs)	Random (pairs)
100	966	774
90-100	20	151
80-90	11	49
< 80	3	26

It is obvious that the accuracy for compound pairs from pathways is higher than that for randomly selected compound pairs when $R_{max} = 15,000$ and $S_{min} = 2$. Since chemical compounds are modified little by little along pathways, the compound pairs along pathways usually exhibit high similarity scores. Namely, the compound pairs from pathways are closely related and such closely related pairs can be found rapidly with our heuristics. In contrast, an optimal solution for random pairs is likely to consist of small multiple matched regions, which apparently require much longer time to compute as indicated by the larger differences between the solutions with R_{max} = 15,000 and 10,000,000. Thus, we conclude that the heuristics of R_{max} and S_{min} work well to identify closely related compounds with biochemical significance with a reasonable amount of time.

4.5 Availability of Software

All programs in the SIMCOMP are written in C language and Perl script language and intended to work well on most of standard UNIX operating systems. The source codes are available from our web site `http://web.kuicr.kyoto-u.ac.jp/simcomp/`. One can find hardware and software requirements and detailed instructions for installation of the package.

Acknowledgments

We thank Koichiro Tonomura, Rumiko Yamamoto, Tomoko Komeno, and Masaaki Kotera for checking the compound and reaction data in the course of preparing our dataset. We also thank all of the KEGG project team members for maintaining and updating the LIGAND and PATHWAY databases, without which this work would not have been possible. This work was supported by the grants from the Ministry of Education, Culture, Sports, Science and Technology of Japan, the Japan Society for the Promotion of Science, and the Japan Science and Technology Corporation. All of the computational resources were provided by the Bioinformatics Center, Institute for Chemical Research, Kyoto University.

References

[1] Allen, F.H. and Kennard, O., 3D search and research using the Cambridge structural database, *Chemical Design Automation News*, 8:1&31–37, 1993.

[2] Bron, C. and Kerbosch, J., Algorithm 457: finding all cliques of an undirected graph, *Communications of the ACM*, 16:575–577, 1973.

[3] Brown, R.D. and Martin, Y.C., Use of structure-activity data to compare structure-based clustering methods and descriptors for use in compound selection, *J. Chem. Inf. Comput. Sci.*, 36:572–584, 1996.

[4] Brown, R.D. and Martin, Y.C., The information content of 2D and 3D structural descriptors relevant to ligand-receptor binding, *J. Chem. Inf. Comput. Sci.*, 37:1–9, 1997.

[5] Flower, D.R., On the properties of bit string-based measures of chemical similarity, *J. Chem. Inf. Comput. Sci.*, 38:379–386, 1998.

[6] Forsythe, R.G. Jr., Karp, P.D., and Mavrovouniotis, M.L., Estimation of equilibrium constants using automated group contribution methods, *Comput. Appl. Biosci.*, 13:537–543, 1997.

[7] Goto, S., Nishioka, T., and Kanehisa, M., LIGAND: chemical database for enzyme reactions, *Bioinformatics*, 14:591–599, 1998.

[8] Goto, S., Okuno, Y., Hattori, M., Nishioka, T., and Kanehisa, M., LIGAND: database of chemical compounds and reactions in biological pathways, *Nucleic Acids Res.*, 30:402–404, 2002.

[9] Hattori, M., Okuno, Y., Goto, S., and Kanehisa, M., Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways, *J. Amer. Chem. Soc.*, 125:11853–11865, 2003.

[10] Jaccard, P., The distribution of the flora of the alpine zone, *New Phytologist*, 11:37–50, 1912.

[11] James, C.A., Weininger, D., and Delany, J., *Daylight Theory Manual 4.71*, Daylight Chemical Information Systems, Inc., Irvine, CA., 2000.

[12] Kanehisa, M., A database for post-genome analysis, *Trends Genet.*, 13:375–376, 1997.

[13] Kanehisa, M., Goto, S., Kawashima, S., and Nakaya, A., The KEGG databases at GenomeNet, *Nucleic Acids Res.*, 30:42–46, 2002.

[14] Kanehisa, M. and Bork, P., Bioinformatics in the post-sequence era, *Nature Genetics*, 33:305–310, 2003.

[15] Kuhl, F.S., Crippen, G.M., and Friesen, D.K., A combinatorial algorithm for calculating ligand binding, *J. Comp. Chem.*, 5:24–34, 1984.

[16] Mavrovouniotis, M.L., Group contributions for estimation standard Gibbs energies of formation of biochemical compounds in aqueous solution, *Biotech. Bioeng.*, 36:1070–1082, 1990.

[17] Mavrovouniotis, M.L., Estimation of standard Gibbs energy changes of biotransformations, *J. Biol. Chem.*, 266:14440–14445, 1991.

[18] Miller, M.A., Chemical database techniques in drug discovery, *Nature Rev. Drug Discovery*, 220:220–227, 2002.

[19] Murzin, A.G., Brenner, S.E., Hubbard, T., and Chothia, C., SCOP: a structural classification of proteins database for the investigation of sequences and structures, *J. Mol. Biol.*, 247:536–540, 1995.

[20] Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B., and Thornton, J.M., CATH-A hierarchic classification of protein domain structures, *Structure*, 5:1093–1108, 1997.

[21] Raymond, J.W., Gardiner, E.J., and Willett, P., RASCAL: Calculation of graph similarity using maximum common edge subgraphs, *Comput. J.*, 45:631–644, 2002.

[22] Raymond, J.W., Gardiner, E.J., and Willett, P., Heuristics for similarity searching of chemical graphs using a maximum common edge subgraph algorithm, *J. Chem. Inf. Comput. Sci.*, 42:305–316, 2002.

[23] Raymond, J.W. and Willett, P., Maximum common subgraph isomorphism algorithms for the matching of chemical structures, *J. Comput.-Aided Mol. Des.*, 16:521–533, 2002.

[24] Takahashi, Y., Maeda, S., and Sasaki, S., Automated recognition of common geometrical patterns among a variety of three-dimensional molecular structures, *Analytica Chimica Acta*, 200:363–377, 1987.

[25] Tatusov, R.L., Koonin, E.V., and Lipman, D.J., A genomic perspective on protein families, *Science*, 278:631–637, 1997.

[26] Watson, G.A., An algorithm for the single facility location problem using the Jaccard metric, *SIAM J. Sci. Stat. Comput.*, 4:748–756, 1983.

[27] Weininger, D., SMILES 1. Introduction and Encoding Rules, *J. Chem. Inf. Comput. Sci.*, 28:31–36, 1988.

[28] Willett, P., Winterman, V., and Bawden, D., Implementation of nearest-neighbor searching in an online chemical structure search system, *J. Chem. Inf. Comput. Sci.*, 26:36–41, 1986.

[29] Willett, P., Searching for pharmacophoric patterns in databases of three-dimensional chemical structures, *J. Mol. Recog.*, 8:290–303, 1995.

[30] Willett, P., Barnard, J., and Downs, G.M., Chemical similarity searching, *J. Chem. Inf. Comput. Sci.*, 38:983–996, 1998.