Chemical Substructure Search in SQL

Adel Golovin and Kim Henrick*

EMBL-EBI Hinston Hall Genome Campus, Cambridge, U.K.

Received August 27, 2008

We present a novel technique for a fast chemical substructure search on a relational database by use of a standard SQL query. The symmetry of a query graph is analyzed to give additional constraints. Our method is based on breadth-first search (BFS) algorithms implementation using Relational Database Management Systems (RDBMS). In addition to the chemical search we apply our technique to the field of intermolecular interactions which involves nonplanar graphs and describe how to achieve linear time performance along with the suggestion on how to sufficiently reduce the linear coefficient. From the algorithms theory perspective these results mean that subgraph isomorphism is a polynomial time problem, hence equal problems have the same complexity. The application to subgraph isomorphism in chemical search is available at http://www.ebi.ac.uk/msd-srv/chemsearch and http://www.ebi.ac.uk/msd-srv/msdmotif/chem. The application to the network of molecule interactions is available at http://www.ebi.ac.uk/msd-srv/msdmotif.

INTRODUCTION

Drug discovery is moving from extensive experiments toward computer models. The accuracy of these models is very much dependent on the underlying data and on the algorithms for establishing a lead chemical structure. Similarity search of a lead template on a database of small molecules is a crucial stage in drug design. There are two diverse chemical search strategies - structure similarity search and substructure search. Development of INCHI and INCHI key provided a technique to approach the first problem through the description of automorphic structures. It is utilized by text based chemical search engines with a simplified similarity option by the insertion of wildcard characters. Although it allows a very efficient search, it fails for a lead template because of its substructure nature. Therefore here we address the second problem, substructure matching.

Substructure search methods have been extensively reported since 1950 and include backtracking, Ullmann's, Von Scholley's, hierarchical trees, and others.[1] It resulted in a number of available fast search engines.[2−4] These systems usually implement a combination of methods, where majority, like backtracking, is based on Depth-first Search (DFS) algorithm. The advantage of DFS is that it allows heuristics which in most cases lead to nearly linear time performance. Although, an alternative to DFS, BFS algorithm is usually regarded as cost inefficient for the problem, we achieve our results through the adoption of its revised version.

Here we describe a novel procedure of wrapping a subgraph isomorphism into a standard SQL query without function calls. It takes an advantage of a BSF implementation on relational database engines combined with the mathematics used in the Constraint Satisfaction Problem (CSP). CSP has found a wide range of applications to artificial intelligence, graph theory, and operational research. Any problem where there is a set of objects, each object has a set of states, and these states are constrained or interconnected can be described as a CSP. Here we address a subclass of CSP known as Finite CSP. This subclass can be represented as a relational database[5,6] where we utilize symmetry breaking techniques[7−10] to gain performance.

This approach has a number of advantages, including platform independency, simplicity, flexibility, integrity, robustness, and single point of failure. It transfers the issues of session, transaction, memory, and CPU management onto a RDBMS; therefore, SQL chemical search automatically fits into a local infrastructure where the chemical libraries are held in a database repository.

METHOD

A CSP framework for subgraph isomorphism was proposed by Rudolf[11] along with the suggestion of using a query mechanism. This technique splits the problem into two halves, where the first one is generation of a query and the latter is application of this query to a set of graphs. In our application we use a basic SQL query, which consists of "select", "from", and "where" clauses. An example for O═C−N chemical substructure is shown in Figure 1.

We refer to SQL query on relational database as CSP using the following definition: CSP is a triple $P = <V,D,C>$ where V is a set of variables or tables in the "from" clause, D is the universal domain of variable values which is the physical data in the database tables, and C - "where" clause, is the set of constraints that specifies which assignment of domain values to the variables is a solution.

The query is executed using an RDBMS on a database of chemical compounds. To implement the database the basic tables for molecule graphs are required. These tables are molecules, atoms, and bonds, i.e. a textual descriptor of a molecule with its QSAR attributes, a textual descriptor of the molecules atoms, and the information about chemical bonds. The latter has an extended meaning of the bond-type. This column encapsulates three attributes - elements of the

* Corresponding author phone: +44 (0) 1223 494629; fax: +44(0) 1223 494468; e-mail: henrick@ebi.ac.uk.

10.1021/ci8003013 CCC: $40.75
Published on Web 12/15/2008

```
Select a1.molecule_id, a1.id, a2.id, a3.id
From bonds b1, bonds b2, atoms a1, atoms a2, atoms a3
Where a1.element = 'O' and a2.element = 'C' and a3.element = 'N'
and b1.atom1_id = a1.id and b1.atom2_id = a2.id and b2.atom1_id =
a2.id and b2.atom3_id = a3.id and b1.bond_type = 'DOUBLE' and
b2.bond_type = 'SINGLE'
```

Figure 1. O═C−N substructure search SQL query to a chemical database. This basic query consists of **Select**, **From**, and **Where** clauses. In CSP terms database tables in the **From** clause represent variables, those content, data, represents the domain, and **Where** clause represents constraints.

```
Select b1.compound_id, b1.atom1_id, b1.atom2_id, b2.atom2_id
From bonds b1, bonds b2
Where b1.bond_type = XXX
and b2.atom1_id = b1.atom1_id and b2.bond_type = YYY
```

Execution plan:

Step	Operation	Object used
1	BITMAP INDEX SINGLE VALUE	bond_type index
2	NESTED LOOP with INDEX RANGE SCAN	Atom1_id index

Figure 2. O═C−N substructure search rewritten SQL query and its query plan. In this example bond type XXX means a Carbon atom double bonded to an Oxygen atom, and YYY means a Carbon atom single bonded to a Nitrogen atom. The first table is accessed via a bitmap index on bond_type column and the second table is accessed by the adjusted list index on atom1_id column. The index range scan operation automatically sorts results in the order of the index.

Table 1. Demonstrates That Our Implementation of BFS on a Relational Database Performs a Search of the Given Substructures in Linear Time[a]

substructure	time (ss)	chemical compds
C−C	7.46	8296
C−C−C	10.80	7761
C−C−C-C	18.77	6541
C−C−C−C−C	18.76	5339

[a] The five Carbon atoms chain search takes the same time as four atoms chain due to reduction in the number of hits. The search database consists of PDB chemical compounds with about 8000 molecules installed on a Sun T2000 computer with Oracle 9i (9.2.0.6.0) RDBMS. It is available at http://www.ebi.ac.uk/msd-srv/msdmotif/chem.

connected atoms pair and the bond order (single, double, triple,...). Introduction of the elements pair in addition to the bond order increases cardinality of the column values by ∼1000 depending on the number of unique pairs of general elements for which a bond exists in the given library of chemical compounds. It allows writing of queries over the bonds table only. In addition to these tables we require three indexes on the bond table. The first index is bond type. The second index is the unique identifier of the first atom (by atom1_id column). This index converts the bond table into an adjusted list for each atom giving all its neighbors. An adjusted list is an advantageous data representation for searches made on a sparse graph, thus matching the topology graph typically observed for chemical compounds. The third index is a unique identifier of records in the bonds table by atom pairs.

Composition of the SQL query given a substructure is accomplished by building a spanning tree, where the rarest tree is chosen from all alternatives. The choice is based on the database statistics. Such a tree complies with the CSP principle that variables with the highest probability of failure go first. The first failure principle terminates search at the i - 1 edge where "i" is the order of the graph's edge at which the failure occurred. This saves generally M^{N-i} number of operations from being performed by the search engine[11] (here M is an average branching in a vertex i.e. an average number of direct neighbors - 1, N is the number of edges). The tree is used to form the SQL query where the tables in the "from" clause follow the order of the tree from the root to the leaves. Where the RDBMS permits index usage assignment, the indexes are listed as follows: for the root to use bond-type index, for tree branches, and for leaves the adjusted list index is recommended and for the closing loop edges the primary key index is in use, hence these edges are calculated in linear time.

With this respect the query in Figure 1 can be rewritten, and its query plan can be viewed as shown in Figure 2.

As can be seen from the query plan the database engine performs filtering on the first step and then resolves the subgraph isomorphism problem using the second index. On the latter step it implements a general nested loop access, which is literally BFS, to resolve the second bond. The feature of this step is the second index usage and the range index scan implementation. To show it in detail we consider another example with three bonds of the equal type connecting four Carbon atoms: C−C−C−C.

The SQL query is

Select b1.atom1_id, b2.atom1_id, b3.atom1_id
From bonds b1, bonds b2, bonds b3
Where b1.bond_type = ZZZ
and b2.atom1_id = b1.atom2_id and b2.bond_type = ZZZ
and b2.atom2_id != b1.atom1_id
and b3.atom1_id = b2.atom2_id and b3.bond_type = ZZZ
and b3.atom2_id != b2.atom1_id

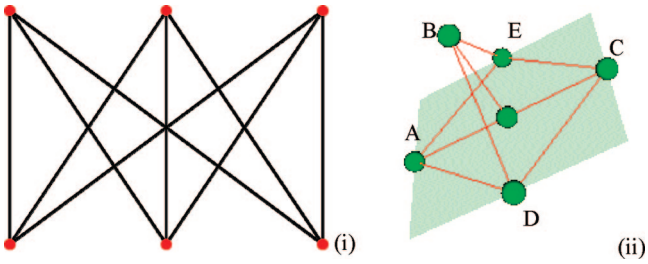

Figure 3. K3,3 graph also known as the utility graph: (i) the classical representation and (ii) the graph drawn to expose the two symmetry planes, ABC and DBE. The graph has 9 edges, where each edge in a vertex connects to 2 others (branching = 2). Therefore the general complexity of the graph isomorphism problem is $2^9 = 512$; however, 4 edges of the graph do close the loops (for instance the AD edge closes the ACD loop, the AE edge closes ACE loop), these edges can be resolved in linear time and therefore are readily removed. Due to the 2 planes of symmetry, symmetry breaking decreases the complexity by a degree of 2, hence the exponential part of the problem's complexity is $2^{9-4}/2^2 = 8$, which is less than the number of edges (9).

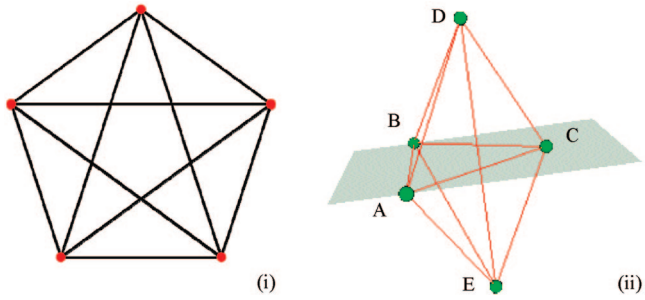

Figure 4. K5 graph: (i) the classical representation and (ii) the graph drawn to expose the four symmetry planes, ABC,CDE, ADE, and BED. The graph has 10 edges, where each edge in a vertex connects to 3 others (branching = 3); 6 of the edges do close the loops. The exponential part of the problem complexity is $3^{10-6}/3^4 = 3$ which is less than the number of edges (10).

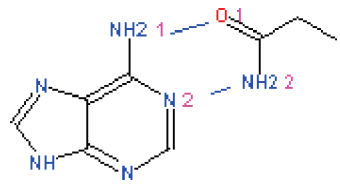

Figure 5. Bidentate interaction between adenine with amide groups.

and b3.atom2_id != b1.atom1_id

Here ZZZ is an identifier of a single bond between two Carbon atoms.

Assuming that a database has N carbon−carbon bonds and the average connectivity of a carbon atom is 3, hence its branching is 2. A general BFS would perform N operations on the first step, N×2 operations on the second step, and N×2×2 operations on the third step. The number of operations rises exponentially with the number of bonds; however, relational databases allow an optimized BFS implementation, which operates on sets. In the example each step maps a set of N Carbon−Carbon single bonds to itself, and there are about N×2 possible mappings on each step. This is achieved by the use of the index range scan access to each next table which automatically sorts results in the order of the index. It is important to note that there is no expansion from one step to another. Each mapping is stored and is used at the last step when the database engine performs a trace back of the hits. Most of the queries can be resolved in linear time. Table 1 provides examples of such queries against a PDB chemical compounds database.

However the last step of the algorithm, where the hits are traced backward, can have an exponential cost, e.g., when the mappings on previous steps map each atom to all others (e.g., - complete graph) then there will be N×N×N hits and to retrieve them from the database the search engine will need to perform an exponential number of operations. We address it by applying a specific to the problem knowledge: the hits must differ by at least one bond and the sequence of the bonds can be any. Therefore the number of the hits is limited to N and all other hits are symmetry related. To resolve the symmetry problem we incorporated a symmetry breaking technique.

CSP detection of symmetry in the variable is the same as the graph isomorphism problem and often regarded as NP-complete though without a known proof. We approached this problem by using a heuristic[7] but for the detection of both global and local bilateral (mirror) and radial (central) symmetries. The elimination of graph symmetry in the SQL can be done using less (<) or more (>) operands instead of the vertices nonequality: != (*all different* in terms of CSP). Symmetry breaking gives the order of the vertices and thus labels the graph, hence exponentially reducing the number of operations by the degree of the ordered vertices number. The complete elimination of the symmetry ensures that all vertices in the subgraph are ordered and therefore leads to a linear time algorithm. However, the number of constraints to remove the symmetry can be exponential itself.[9] A complete graph with equal vertices and edges is an example of the graph with an exponential number of symmetries. In such cases, special care, like choosing a path on the graph, is taken to apply only the necessary number of constraints, i.e.: N-1.

Complete graphs with five vertices or more are nonplanar graphs. The nonplanar graphs are those which cannot be drawn in 2D without edge intersections. Present graph theory does not give a satisfactory answer with regards to the complexity of the subgraph isomorphism problem for nonplanar graphs. Although the vast majority of chemical compounds can be represented as planar graphs, when consideration of a network of molecular interactions involving ionic, hydrogen, and van der Waals bonds is taken into account, the result is mainly nonplanar graphs. For the nonplanar graphs, we suggest that symmetry breaking will give a new insight into the isomorphism problem. Taking into account that linear time algorithms exist for planar graphs[12] and in accordance with Kuratowski's reduction theorem,[13] nonplanar graphs must have K3,3 or K5 as the smallest subgraph. To investigate that a linear time algorithm exists for nonplanar graphs we started with symmetry breaking in K3,3 and K5. However, as shown in Figures 3 and 4, K3,3 and K5 graphs do expose explicit bilateral symmetry, hence our first result is that the planarity analysis must be a linear time problem. In accordance with the theorem, that any graph can be presented as a composition of planar, K3,3 and K5 graphs then the algorithm for this presentation is at most an N square problem. This composition can consequently be presented as a graph where the nodes are subgraphs and the edges are the edges that connect these subgraphs. Such representation reduces the number of

Table 2. Search Examples against 931,007 Chemical Compounds on a Sun T2000 Computer with Oracle 9i (9.2.0.6.0)[a]

	strict substructure		discard bond order		strict ring/no ring	
SMILES	time (s)	hits	time (s)	hits	time (s)	hits
ONC1CC(C(O)C1O)[n]2cnc3c(NC4CC4)ncnc23 adenosine derivative	1.35	1	1.37	1	1.54	1
Nc1ncnc2[nH]cnc12 - adenine	8.28	2752	17.77	2923	1.64	2742
CNc1ncnc2[n](C)cnc12 - adenine derivative	1.84	541	4.73	582	1.48	510
Nc1ncnc2[n](cnc12)C3CCCC3 - adenine derivative	1.26	86	1.86	115	1.26	84
CC12CCC3C(CCC4=CC(O)CCC34C)C1CCC2 - steroid	5.06	73	23.12	6826	3.17	71
CC12CCCC1C3CCc4 cm^3(O)ccc4C3CC2 - steroid	3.18	632	20.51	7909	9.29	630
OC2=CC(=O)C1=C(C=CC=C1)O2 - oxochromen	3.54	16	40.33	1074	3.49	3
NC1=NN=C(S)S1 1,3,4 thiazole	1.28	1071	1.40	1215	1.25	1047
C1C2SCCN2C1 - penicillin fragment	1.30	148	1.29	149	1.16	148
CP(O)(O)=O - oxophosphorane	3.69	3160	1.50	3194	1.28	3027
CCCCCP(O)(O)=O - oxophosphorane lipid	1.36	300	1.52	1681	1.29	157
N2CCC13CCCCC1C2CC4=C3C=CC=C4 - alkaloid	1.90	150	3.13	306	1.28	150
S1C=NC=C1 - thiazole	36.34	28371	36.93	70353	36.35	28371
C34CCC1C(CCC2CC(=O)CCC12)C3CCC4 - steroid	1.65	462	29.75	8024	1.92	462
CCCCCCCCCCCP(O)(O)=O oxophosphorane lipid	1.33	72	1.49	171	1.36	45

[a] Another test using Sun Fire X4600 with Oracle 11g on Linux showed improvement of the performance by an order of magnitude. A threshold of 1.24 s has been built into the Web application, and a less response time cannot be observed. The general tendency is that the less specific query, the more hits it gets, the more time it takes to retrieve and sort them. Hence large substructures are executed faster than small. Another reason for the difference in the execution time is the use of the first failure principle which generally picks up a most rare bond as the first in the query.

vertices in the graph and iteratively converges to a planar, K3,3 or K5 graphs. Therefore graph isomorphism is a linear composition of linear time problems, hence, itself is at most an N square problem.

By solving the graph isomorphism problem for a query graph in N square times, we have an extended query with the symmetry breaking constraints for resolving the subgraph isomorphism problem on a large data set in linear time.

The efficient application to chemical searches requires prior chemical knowledge with special attention paid to the assignment of bond aromaticity. We have implemented the definition of aromatic bonds as described in refs 14 and 15.

For faster search the FREL[1,16] (Fragment Reduced to an Environment which is Limited) technique with two levels was incorporated by the use of database indexes. An additional filtering by rare bonds and rings is used as a first statement in the query too.

RESULTS

The method was tested with a database of 931 007 chemical compounds from several public sources. These included the Open NCI database[17] and a number of vendor catalogs of available chemicals. The set contains ap-

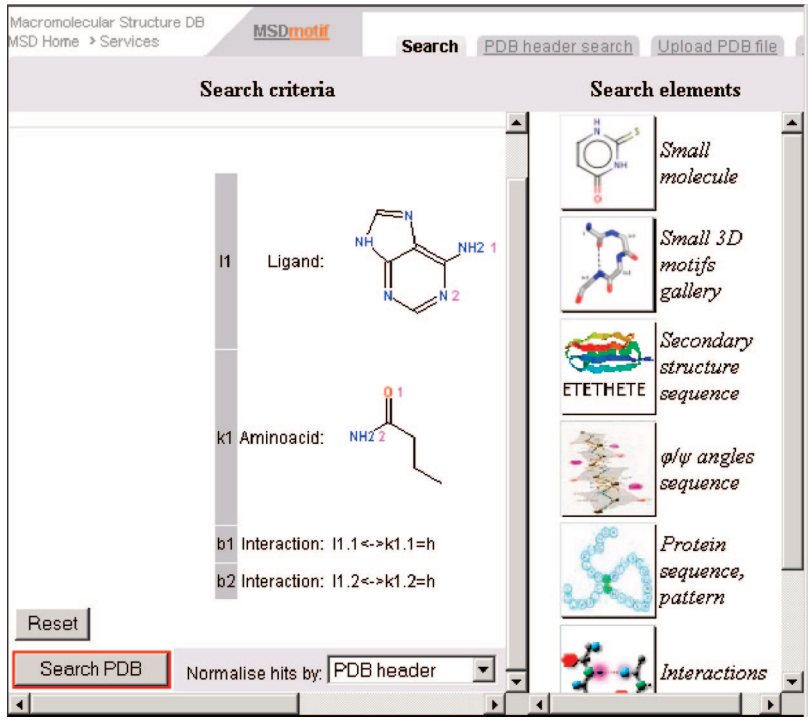

Figure 6. MSDmotif interface with interactions specified between an amide group of a protein and the adenine group.

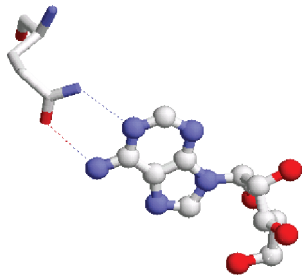

Figure 7. 1bx4:[22] PDB entry where the adenosine group has a glutamine 289 interaction.

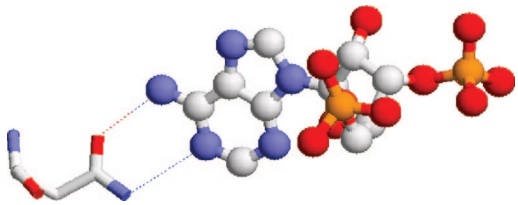

Figure 8. 1o0f: [23] PDB entry where the adenosine-3′,5′-diphosphate group has an asparagine 71 interaction.

proximately 250 000 compounds with 3D coordinates and the rest with 2D coordinates. It is accessible via http://www.ebi.ac.uk/msd-srv/chemsearch. The database contains a nonredundant set where the normalization was done with respect to stereo SMILES. The loading of the data into the Oracle 9i database on a Sun T2000 machine together with indexing takes about 8 h.

The search options include strict substructure search, with the possibility of discarding single/double bond order, which allows for flexibility where single and double bonds are treated as equal, and an option to search with strict ring and nonring matching of atoms to the search substructure. The latter option is particularly useful in searching for lipids.

Examples of substructure searches against this set are provided in Table 2.

A second service at http://www.ebi.ac.uk/msd-srv/msd-motif/chem is provided to access ∼8000 chemical compounds from the Protein Data Bank (PDB).[18]

A third service using this method, MSDmotif,[19] is available at http://www.ebi.ac.uk/msd-srv/msdmotif. It is based on the Macromolecular Structure Database (MSD)[20] which contains ligand-protein interactions annotated as described in MSDsite.[21] MSDsite provides a ligand environment search for PDB from a given ligand and amino-acid three letters codes, where the environment has a star architecture with the ligand as the "sun". In MSDmotif integrated search engine we have extended this functionality by the use of substructure searches where polymer−ligand interactions may be queried. This is illustrated by the example of finding bidentate H-bonded fragments in the PDB between an adenine group and a GLN or ASN (see Figure 5) side chain.

The input is by drawing each substructure using the Java Molecule Editor (JME), by labeling NH2-1, N - 2 of adenine along with O - 1, N - 2 atoms of the amide group, and connecting the two graphs by hydrogen bonds using the interface. The MSDmotif interface with this query is shown in Figure 6.

The service automatically generates an SQL query and gives the list of molecule pairs (on ∼50 000 PDB entries in ∼8 s). Further visualization and 3D alignment is possible. For comparison, Relibase[2] at http://relibase.ccdc.cam.ac.uk performs the same search on the same data set in 385 s.

Examples of hits are shown in Figures 7 and 8.

The MSD database includes all the NMR models and includes the estimated quaternary structures (PQS[24]) which gives the equivalent of ∼150 000 coordinate entries with some 400 million atom coordinates.

The above example of adenine-protein interactions may be considered simple and obvious in the field of structural biology. However it represents a formidable task when one generalizes such queries for efficient and rapid searching. Such an example demonstrates the power and efficiency of the method we present here, where it is possible to identify modes of interactions that can be further used in drug discovery fields.

ACKNOWLEDGMENT

MSDmotif service is part of work package 13 of the BioSapiens project. The BioSapiens project is funded by the European Commission within its FP6 Programme, under the thematic area "Life sciences, genomics and biotechnology for health", Contract No. LHSG-CT-2003-503265.

REFERENCES AND NOTES

(1) Barnard, J. Substructure Searching Methods: Old and New. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 532–538.
(2) Hendlich, M.; Bergner, A.; Gunther, J.; Klebe, G. Relibase: design and development of a database for comprehensive analysis of protein-ligand interactions. *J. Mol. Biol.* **2003**, *326*, 607–620.
(3) Klekota, J.; Roth, F. P.; Schreiber, S. L. Query Chem: a Google-powered web search combining text and chemical structures. *Bioinformatics* **2006**, *22*, 1670–1673.
(4) Chemaxon. JChem cartridge. http://www.chemaxon.com/product/jc_cart.html (accessed Oct 31, 2008).
(5) Zhang, Y.; Mackworth, A. K. In *Parallel and Distributed Algorithms for finite Constraint Satisfaction Problems*;Proceedings of the 3rd IEEE Symposium In Parallel and Distributed Processing, Dallas, TX, 1991; IEEE: Dallas, TX, 1991; pp 394−397.
(6) Gyssens, M.; Jeanovs, P. G.; Cohen, D. A. Decomposing CSP using database technique. *Artif. Intell.* **1994**, *1* (66), 57–89.
(7) Contreras, M. L.; Alvarez, J.; Guajardo, D.; Rozas, R. Understanding Topological Symmetry: A Heuristic Approach to Its Determination. *J. Comput. Chem.* **2007**, *29* (4), 588–600.
(8) Crawford, J. In *A theoretical analysis of reasoning by symmetry in first-order logic*: Proceedings of AAAI-92 workshop on tractable reasoning, San Jose, CA, 1992; Crawford, J., Ed.; The AAAI Press: CA, pp 17−22.
(9) Crawford, J.; Ginsberg, M.; Luks, E.; Roy, A. In *Symmetry-Breaking Predicates for Search Problems*; Proceedings of Fifth International Conference on Principles of Knowledge Representation and Reasoning, Cambridge, MA, 1996; Morgan Kaufmann Publishers: San-Francisco, CA, 1996.
(10) Flener, P.; Pearson, J.; Sellmann,M.; Hentenryck,P. V.; Agren M. In *Efficient Structural Symmetry Breaking for Constraint Satisfaction Problems*; Proceedings of the International Symmetry Conference, Edinburgh, U.K., Jan 2007; Gen, I. P., Linton, S., Eds.; ISC: Edinburgh, 2007.
(11) Rudolf, M. Utilizing constraint satisfaction techniques for efficient graph pattern matching. *Lect. Notes Comput. Sci.* **1998**, *1764*, 238–251.
(12) Hopcroft, J. E.; Wong, J. K. *In Linear time algorithm for isomorphism of planar graphs (preliminary report)*; Proceedings of Conference Record of Sixth Annual ACM Symposium on Theory of Computing, Seattle, WA, U.S.A., 1974; ACM: New York, NY, 1974; pp 172−184.
(13) Thomassen, C. Kuratowski's theorem. *J. Graph Theory* **1981**, *5* (3), 225–241.
(14) Schleyer, P. v. R. Aromaticity (Editorial). *Chem. Rev.* **2001**, *101*, 1115–1118.
(15) Balaban, T.; Schleyer, P. v. R.; Rzepa, H. S. Crocker, Not Armit and Robinson, Begat the Six Aromatic Electrons. *Chem. Rev.* **2005**, *105*, 3436–3447.

(16) Barth, A.; Westermann, U; Pasucha, B. Messenger and S4: A Comparison of Structure Search Systems. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 714–722.
(17) Voigt, J. H.; Bienfait, B.; Wang, S.; Niclaus, M. C. Comparison of the NCI Open Database with Seven Large Chemical Structural Databases. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 702–712.
(18) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.
(19) Golovin, A.; Henrick, K. MSDmotif: exploring protein sites and motifs. *BMC Bioinformat.* **2008**, *9*, 312.
(20) Boutselakis, H.; Copeland, J.; Dimitropoulos, D.; Fillon, J.; Golovin, A.; Henrick, K.; Hussain, A.; Ionides, J.; John, M.; Keller, P.; Krissnel, E.; McNeil, P.; Naim, A.; Newman, R.; Oldfield, T.; Pineda, J.; Rachedi, A.; Sitnov, A.; Sobhany, S.; Suarez-Uruena, A.; Swaminathan, J.; Tagari, M.; Tate, J.; Tromm, S.; Velankar, S.; Vranken, W. E-MSD: the European Bioinformatics Institute Macromolecular Structure Database. *Nucleic Acids Res.* **2003**, *31*, 458–462.
(21) Golovin, A.; Dimitropoulos, D.; Oldfield, T.; Rachedi, A.; Henrick, K.; MSDsite, A. Database Search and Retrieval System for the Analysis and Viewing of Bound Ligands and Active Sites. *Proteins: Struct., Funct., Bioinf.* **2005**, *58* (1), 190–199.
(22) Mathews, I. I.; Erion, M. D.; Ealick, S. E. Structure of human adenosine kinase at 1.5 A resolution. *Biochemistry* **1998**, *37*, 15607–15620.
(23) Leonidas, D. D.; Chavali, G. B.; Oikonomakos, N. G.; Chrysina, E. D.; Kosmopoulou, M. N.; Vlassi, M.; Frankling, C.; Acharya, K. R. High-resolution crystal structures of ribonuclease A complexed with adenylic and uridylic nucleotide inhibitors. Implications for structure-based design of ribonucleolytic inhibitors. *Protein Sci.* **2003**, *12*, 2559–74.
(24) Henrick, K.; Thornton, J. PQS: a protein quaternary structure file server. *Trends Biochem. Sci.* **1998**, *23*, 358–361.

CI8003013