BIOINFORMATICS

Vol. 18 no. 3 2002
Pages 465–469

The complexity of comparing reaction systems

Mark Ettinger

Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received on August 28, 2001; revised on October 30, 2001; accepted on November 9, 2001

ABSTRACT

Motivation: As more genomic data becomes available there is increased attention on understanding the mechanisms encoded in the genome. New XML dialects like CellML and Systems Biology Markup Language (SBML) are being developed to describe biological networks of all types. In the absence of detailed kinetic information for these networks, stoichiometric data is an especially valuable source of information. Network databases are the next logical step beyond storing purely genomic information. Just as comparison of entries in genomic databases has been a vital algorithmic problem through the course of the sequencing project, comparison of networks in network databases will be a crucial problem as we seek to integrate higher-order network knowledge.

Results: We show that comparing the stoichiometric structure of two reactions systems is equivalent to the graph isomorphism problem. This is encouraging because graph isomorphism is, in practice, a tractable problem using heuristics. The analogous problem of searching for a subsystem of a reaction system is NP-complete. We also discuss heuristic issues in implementations for practical comparison of stoichiometric matrices.

Contact: ettinger@lanl.gov

1 INTRODUCTION

Enormous quantities of genomic data are now available for a wide variety of organisms. It is widely recognized that deciphering the genetic circuitry encoded in the raw sequence data will be as difficult, if not more so, than obtaining the sequences themselves. Clearly computational methods will be a powerful tool in the endeavor to understand the biological networks that govern life. In addition, we will require new languages to describe these biological systems, databases to store these descriptions, and algorithms for comparing this higher-order information. New XML dialects like Systems Biology Markup Language (SBML; Hucka *et al.*, 2001) and CellML (Hedley and Nelson, 2001) were created to provide a language capable of precisely expressing the structure of biological systems.

The work in this paper is motivated by the idea that there will soon be comprehensive databases of reactive systems of the sort which can be specified using SBML and CellML. Basically, we consider reactive systems exemplified by the following:

$$X + Y \longrightarrow Z \quad (1)$$

$$A \longrightarrow B + Y \quad (2)$$

$$B + 2C \longrightarrow X. \quad (3)$$

Searching these databases for a specified reaction system and comparing reaction systems will be as common as genomic sequence searches are now. For example, the above reaction system is actually the same (isomorphic), in a sense to be defined, as the following:

$$T + 2C \longrightarrow D$$
$$D + U \longrightarrow R$$
$$P \longrightarrow T + U.$$

In the second reaction system we have simply renamed the variables and permuted the reactions. How do we automatically recognize such an identity and how hard is such a recognition problem?

The present result characterizes the computational complexity of several of these search and comparison problems for reaction systems. We only consider *syntactic* comparisons between reaction systems because the issue of comparing *dynamics* seems much more difficult. For this reason we focus on the stoichiometry of the reaction systems and ignore the kinetic element. The stoichiometric structure of reaction systems can be represented by a matrix (Heinrich and Schuster, 1996) where the rows are elements entering into the equations, columns are reactions, and entries represent the stoichiometric coefficients. For example, the above reaction system has the stoichiometric matrix

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & -2 \end{pmatrix}$$

where the variables represented by the rows are, from top to bottom, X, Y, Z, A, B, C and the columns from left to

right represent reactions 1–3. For example, the top entry in the first column is -1 because one molecule of X is consumed in the first reaction of the system. Therefore the comparison of reaction systems reduces to the comparison of matrices. We show that the problem of deciding whether two identically sized matrices are isomorphic (i.e. whether two reaction systems with the same number of reactions and the same number of variables) is equivalent to deciding if two graphs are isomorphic. We also point out that the equally important problem of deciding when one matrix is a submatrix of another (i.e. when one reaction system is a subreaction system of another) is NP-complete. We remark that the *Graph Isomorphism* (GI) problem is a very important problem in computer science. It arises in many diverse practical situations. Furthermore, it is one of the few problems for which no polynomial time solution is known nor is it known to be NP-complete. For an overview of the GI problem see Fortin (1996). In practice, GI is usually easy to solve. Efficient heuristic decision procedures rely on exploiting heterogeneities in the graphs to narrow the search space for an isomorphism. We conclude by outlining such an heuristic algorithm for the special matrices arising from reactions systems.

2 COMPLEXITY OF MATRIX AND GRAPH ISOMORPHISM

We are interested in the complexity of comparing matrices. This is a generalization of the problem of comparing graphs. A *graph* is simply a set of vertices and set of edges, where an edge is formally an unordered pair of vertices, $G = (V_G, E_G)$. A *directed graph* is similar except the edges are *ordered* pairs of vertices which indicates a direction to each edge. A *bipartite graph* is a graph where the set of vertices can be divided into two disjoint subsets, $V_G = V_1 \cup V_2$, such that each edge is incident to precisely one vertex in each subset, i.e. if $\{u, v\} \in E_G$ then $u \in V_1$ and $v \in V_2$ or $u \in V_2$ and $v \in V_1$.

The GI problem is to identify two graphs which are, in effect, the same. Formally an isomorphism from G to H is a bijection $f : V_G \to V_H$ such that $\{f(v_1), f(v_2)\} \in E_H$ if and only if $\{v_1, v_2\} \in E_G$. GI is an important, practical problem which arises in many contexts (Fortin, 1996). Notice that GI is clearly in the class NP (Papadimitriou, 1994) as a candidate isomorphism can be easily checked in no greater than $O(|V_G|^2)$ time by comparing all possible edges. It is an unusual problem from the point of view of complexity theory in that most problems which are known to be in NP are known to be in P or to be NP-complete. Neither result is known for GI. A significantly more difficult problem is *subgraph isomorphism* where the map f is only required to be an injection. Subgraph isomorphism is NP-complete. Any problem that is equivalent to GI is called *GI-complete*.

Any graph can be transformed into a bipartite graph as follows. For a graph G let G' denote the bipartite graph obtained by replacing each edge of G with two edges joined by a new vertex. Notice that G and H are isomorphic if and only if G' and H' are isomorphic. This shows that bipartite GI is GI-complete. Furthermore, it shows that bipartite subgraph isomorphism is NP-complete.

DEFINITION 1. Let M and N be matrices with p rows and q columns with entries over the integers m_{ij} and n_{ij}. The *Matrix Isomorphism* (MI) problem is to determine if there exists an isomorphism between the matrices, i.e. a permutation of the rows of M, σ_r and a permutation of the columns of M, σ_c, such that $m_{\sigma_r(i)\sigma_c(j)} = n_{ij}$.

DEFINITION 2. Let M be a $p \times q$ matrix and N a $r \times s$ matrix, $p \leqslant r$ and $q \leqslant s$, both with entries over the integers m_{ij} and n_{kl}. The *SubMatrix Isomorphism* (SMI) problem is to determine if there exists a submatrix of N which is isomorphic to M, i.e. maps f and g such that $m_{ij} = n_{f(i)g(j)}$.

We note the relationship of the above problems with graph-theoretic problems. An arbitrary binary matrix can be regarded as an adjacency matrix of a bipartite graph. Therefore MI generalizes bipartite GI and is therefore at least as hard as GI. SMI generalizes bipartite subgraph isomorphism and since SMI is clearly in NP we see that SMI is NP-complete.

MI is the computational problem which is important for comparing stoichiometric matrices. However for the sake of both theoretical completeness and also future ease of exposition in our proof that MI is GI-complete, we now define a generalization of MI. The idea of the generalization is that the entries of the two matrices, rather than being integers, might originate from two distinct, finite symbol sets. So in addition to finding appropriate row and column permutations, we must also find an appropriate symbol permutation. We note that this is not a *biologically* motivated generalization as the actual stoichiometric coefficients are the defining biological characteristic of the matrices, and thus do not admit a sensible permutation.

DEFINITION 3. Let M and N be matrices with p rows and q columns with entries m_{ij} and n_{ij} over the symbol sets S and T. The *Extended Matrix Isomorphism* (EMI) problem is to determine if there exists an *extended* isomorphism between the matrices, i.e. a permutation of the rows of M, σ_r, a permutation of the columns of M, σ_c, and a bijection of the symbol sets $f : S \to T$ such that $f(m_{\sigma_r(i)\sigma_c(j)}) = n_{ij}$.

THEOREM 1. *EMI is reducible to GI* (*and therefore is GI-complete*).

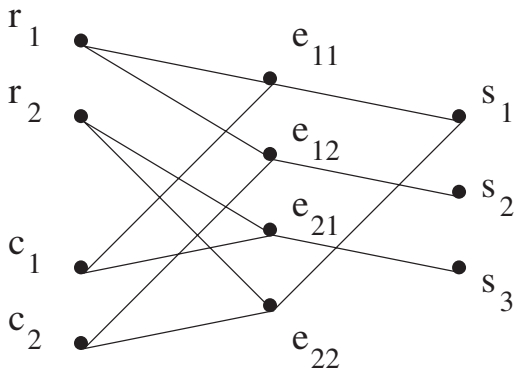

Fig. 1. The graph corresponding to the example matrix M. For a stoichiometric matrix the row vertices would correspond to metabolites, the column vertices to reactions, and the symbol vertices to the stoichiometric coefficients in the reactions. The entry vertices are a mathematical artifact of the construction and have no biological meaning.

PROOF. Assume that M and N are matrices of size $p \times q$, with entries m_{ij}, n_{ij}, over r distinct entries. We now associate graphs G and G' with M and N by the following construction. Let G have the vertices $V = R \cup C \cup E \cup S$, where these are sets of *row, column, entry, and symbol* vertices respectively, with sizes $p, q, p \times q, r$ and elements $\{r_1, \ldots, r_p\}$, $\{c_1, \ldots, c_q\}$, $\{e_{1,1}, \ldots, e_{p,q}\}$, $\{s_1, \ldots, s_r\}$. Let G' have the vertices $V' = R' \cup C' \cup E' \cup S'$, etc. Vertex r_i is connected to all $e_{i,j}$, c_j is connected to all $e_{i,j}$, and $e_{i,j}$ is connected to s_k if and only if $m_{ij} = k$. So each edge vertex is connected to the corresponding symbol vertex as specified in the matrix. Also each row vertex is connected to all the edge vertices with the same row index and similarly for column vertices. We therefore reflect all the structure of the matrix in the corresponding graph. For example, the graph G resulting from $M = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix}$ is illustrated in the figure.

We claim M and N are extended-isomorphic if and only if G and G' are isomorphic. If M and N are extended-isomorphic then they are effectively the same matrix up to row and column permutations and a symbol bijection. Thus the above construction will preserve this and G and G' will be isomorphic. Let h be an isomorphism between the graphs. We may assume that h preserves the row, column, edge, and symbol vertices because we can enforce this by tagging these sets with suitable, distinct labels. It is easy to verify that the following mapping defines an isomorphism from M to N: $\sigma_r(i) = i'$ if and only if $h(r_i) = r'_{i'}$, $\sigma_c(j) = j'$ if and only if $h(c_j) = c'_{j'}$, and $f(s_i) = s'_{i'}$ if and only if $h(s_i) = s_{i'}$. The edges between edge and symbol vertices insures that the entries match up and the edges to the row and column vertices insure that the rows and columns are mapped together as appropriate units. □

By attaching suitable unique labels to the symbol vertices we can insure that any extended isomorphism must fix the symbols. Thus we obtain the following.

COROLLARY 1. *MI is reducible to GI and therefore is GI-complete.*

We have mentioned that SMI is NP-complete. However the biologically relevant problem is less general than SMI. For example, we would like to be able to recognize when one reactive system is a reactive subsystem of a larger system, as in the following:

$$\begin{aligned} X + Y &\longrightarrow Z \\ A &\longrightarrow B + Y \\ B + 2C &\longrightarrow X \\ T + 2C &\longrightarrow D \\ U &\longrightarrow D + P \\ D + U &\longrightarrow R \\ 2R &\longrightarrow T + D \\ P &\longrightarrow T + U. \end{aligned}$$

Notice that when looking for a subsystem, we are only interested in finding an injection of reactions and a *bijection* of reactants. It is not biologically meaningful to eliminate reactants from a reaction. For example we do *not* want to consider

$$X \longrightarrow A$$

to be a subsystem of

$$X + Y \longrightarrow A + B$$

which would be the case under our current formulation of SMI. The stoichiometric matrix for the first reaction is $(-1 \quad 1)^t$ and for the second is $(-1 \quad -1 \quad 1 \quad 1)^t$. Formally the first matrix is a submatrix of the second by eliminating the second and fourth rows, which corresponds to eliminating the reactants Y and B from the reaction. This motivates the following definition of the *Row-restricted SubMatrix Isomorphism* (RSMI) problem.

DEFINITION 4. Let M be a $p \times q$ matrix and N a $r \times s$ matrix, $p \leqslant r$, $q \leqslant s$, both with entries over the integers m_{ij} and n_{kl}. The RSMI problem is to determine if there exists a row-restricted submatrix of N which is isomorphic to M, i.e. maps f and g such that $m_{ij} = n_{f(i)g(j)}$, subject to the restriction that if l is in the image of g, then for all nonzero n_{kl}, k is in the image of f.

Clearly GI reduces to RSMI and RSMI is in NP so RSMI reduces to SMI. Currently, we do not know if RSMI is NP-complete.

3 IMPLEMENTATION

There are two main approaches to solving GI problems in practice. Both methods can be adapted to the MI. The first, attaching a canonical labelling to a graph which uniquely identifies its isomorphism class, applies only to GI, not to SMI. This approach is used in the popular program *nauty* (McKay, 1981). To solve MI one could utilize the reductions in the previous proof and apply *nauty* to the graphs G and G'. See Fortin (1996) for a discussion of why this technique is often more efficient than a direct search for an isomorphism between two graphs. It would be interesting to see how this technique compared in efficiency to our direct approach for a MI outlined below.

The second method, a direct search for an isomorphism utilizing vertex invariants to prune the search tree, is applicable to both GI and SMI. Our heuristic algorithm falls into this category. We suspect that it will perform well on real reactive systems due to the heterogeneities in real stoichiometric matrices. Our algorithm exploits these heterogeneities in its use of row and column invariants.

A vertex invariant is any function i on vertices such that if f is an isomorphism from G to H then $i(v) = i(f(v))$. An example is the degree of a vertex. If $i(v) \neq i(v')$ then there cannot exist an isomorphism which maps v to v'. In this way vertex invariants allow one to narrow the space of candidate isomorphisms to check.

Rather than vertex invariants, our matrices will have row and column invariants. This will restrict which rows can be mapped to which rows under an isomorphism and which columns can be mapped to which columns. Our row and column invariants will be the composition of three subinvariants. Let m_j denote the jth column of matrix M and m_i^t denote the ith row. Then the *type* of m_j, $type(m_j)$, will be the unique, ordered, nonincreasing rearrangement of m_j. For example $type(0 \;\; 1 \;\; -1 \;\; -1)^t) = (1 \;\; 0 \;\; -1 \;\; -1)^t$. The *2-neighborhood* of m_j, $n(m_j)$, is the number of columns that share a reactant with m_j, i.e. $n_2(m_j) = |\{l : \exists i m_{ij} \neq 0, m_{il} \neq 0\}|$. The *3-neighborhood* of m_j, $n_3(m_j)$, is the number of rows that have a nonzero entry in any reaction in the 2-neighborhood of m_j, i.e. $n_3(m_j) = |\{l : \exists i m_i \in n_2(m_j), m_{li} \neq 0\}|$.

We use the same terminology for row invariants. If our matrices are binary then we may consider them to be adjacency matrices of bipartite graphs. In this specialized case the type becomes the *degree* of a vertex, the 2-neighborhood becomes the *twopath*, i.e. the number of vertices reachable along a path of length two, and similarly the 3-neighborhood becomes the *threepath*. We define a column invariant to be the composition of the type, the 2-neighborhood, and the 3-neighborhood, $i(m_j) = (type(m_j), n_2(m_j), n_3(m_j))$, and similarly for a row invariant, $i(n_i)$. For larger matrices where the search space is much larger one could add further invariants, for example the general *n-neighborhood*. We perform an exhaustive search consistent with these invariants. Furthermore we also narrow the search space by utilizing knowledge of unique images of rows and columns determined by the invariants. For example, sometimes the invariants are sufficient to determine that the image of column m_j can *only* be n_k. This implies that if m_{ij} is nonzero then row i can only map to a row l such that n_{lk} is nonzero.

Here is an outline for our algorithm.

MATRIX ISOMORPHISM ALGORITHM

(1) Calculate row and column invariants as described above for both matrices.

(2) Create lists, L, of possible row and column images based on invariants.

(3) Store current L, L $\rightarrow$ old-L.

(4) Update lists of possible images L based on rows and columns with unique images.

(5) If L $\neq$ old-L goto 3.

(6) Use recursive backtracking algorithm to perform a depth-first search of tree of partial isomorphisms. At each node of the tree choose the extension from the appropriate list of possible images.

We implemented the algorithm in MATLAB and found that we were able to regularly identify random permutations of the glycolytic pathway and human pyrimidine metabolic network, represented by 20×21 and 73×32 stoichiometric matrices respectively, in several seconds on a desktop PC. See Heinrich and Schuster (1996) for a diagram of glycolysis and (KEGG) for a diagram of the human pyrimidine metabolic network. Clearly the invariants eliminate the vast majority of the potential $20! \times 21!$ and $73! \times 32!$ search spaces.

It seems much more difficult to develop a practical algorithm for RSMI. We may still utilize column types as invariants because in this formulation of the problem we never delete nonzero entries from a column. However we can no longer utilize row types and the neighborhood invariants must be modified to such a degree that they become far less helpful at reducing the search space. In the case of RSMI we may only use the criterion $n(m_j) > n(n_k)$ as a means of eliminating possible images of columns under potential isomorphisms. Developing a practical algorithm for RSMI for realistic stoichiometric matrices is currently under study.

4 DISCUSSION

We have studied the complexity of comparing reaction systems. We have shown that comparing two reaction systems of the same size is equivalent to the GI problem and that searching a reaction system for a given subreaction system is NP-hard. We have introduced an heuristic algorithm for MI and given empirical evidence

that it will perform well in practice on real stoichiometric matrices. The comparisons take a few seconds on a Pentium III, Windows NT computer with 256 K RAM.

Currently we are investigating heuristics for searching for sub-reaction systems. This problem is very important for searching network databases for a given reaction system. Therefore an heuristic approach to this difficult computational problem is highly desirable. One solution is to simply do a brute force search over all possible submatrices and use the heuristic MI algorithm to answer the resulting MI subproblems. This approach would run in time $C_{MI}\binom{r_N}{r_M}\binom{c_N}{c_M} = C_{MI}\frac{r_N!}{r_M!(r_N-r_M)!}\frac{c_N!}{c_M!(c_N-c_M)!}$ where C_{MI} is the running time to check MI for matrix M and the current submatrix of matrix N, r_N is the number of rows of matrix N, c_N the number of columns, and similarly for r_M and c_M. Finally, it would be very interesting to define a meaningful way of comparing the *dynamics* and *behavior* of reaction systems and assess the computational difficulty of this task.

ACKNOWLEDGEMENTS

We gratefully acknowledge Michael Murphy and Adam Cannon for helpful discussions on this problem. This work was performed under the auspices of the Department of Energy under contract to the University of California and was supported by the Molecular Foundations of Pathogenesis project, funded by Laboratory Directed Research and Development at Los Alamos National Laboratory.

REFERENCES

Fortin,S. (1996) The graph isomorphism problem. *Technical Report 96–20*. University of Alberta, Edmonton, Alberta, Canada, citeseer.nj.nec.com/fortin96graph.html.

Hedley,W. and Nelson,M. (2001) CellML specification, www.cellml.org.

Heinrich,R. and Schuster,S. (1996) *Regulation of Cellular Systems*. Kluwer, New York.

Hucka,M., Finney,A., Sauro,H. and Bolouri,H. (2 March 2001) Systems Biology Markup Language (SBML) level 1: structures and facilities for basic model definitions, http://www/cds.caltech.edu/erato/sbml/docs/index.html.

KEGG (2001) Kyoto Encyclopedia of Genes and Genomes, http:www.genome.ad.jpdbget-binwww-bget?path:hsa00240.

Mckay,B. (1981) Practical graph isomorphism. *Congressus Numerantium*, **30**, 45–87.

Papadimitriou,C.H. (1994) *Computational Complexity*. Addison-Wesley, New York.