JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 18, Number 1, 2011
© Mary Ann Liebert, Inc.
Pp. 43–58
DOI: 10.1089/cmb.2009.0216

Computing Atom Mappings for Biochemical Reactions without Subgraph Isomorphism

MARKUS HEINONEN,[1] SAMPSA LAPPALAINEN,[1] TANELI MIELIKÄINEN,[2] and JUHO ROUSU[1]

ABSTRACT

The ability to trace the fate of individual atoms through the metabolic pathways is needed in many applications of systems biology and drug discovery. However, this information is not immediately available from the most common metabolome studies and needs to be separately acquired. Automatic discovery of correspondence of atoms in biochemical reactions is called the "atom mapping problem." We suggest an efficient approach for solving the atom mapping problem exactly—finding mappings of minimum edge edit distance. The algorithm is based on A* search equipped with sophisticated heuristics for pruning the search space. This approach has clear advantages over the commonly used heuristic approach of iterative maximum common subgraph (MCS) algorithm: we explicitly minimize an objective function, and we produce solutions that typically require less manual curation. The two methods are similar in computational resource demands. We compare the performance of the proposed algorithm against several alternatives on data obtained from the KEGG LIGAND and RPAIR databases: greedy search, bi-partite graph matching, and the MCS approach. Our experiments show that alternative approaches often fail in finding mappings with minimum edit distance.

Key words: algorithms, biochemical networks.

1. INTRODUCTION

Metabolic modeling of a cell is a fundamental part of systems biology (Kell, 2004), where structure, properties, and dynamics of cellular and organismal systems are studied. Advances in system level biology will have an impact on the future of medicine and understanding of organisms (Kitano, 2002). Metabolism is modeled with chemical reactions catalyzed by enzymes that process and transform the compounds of the cell to produce energy and building blocks, with additional functions of information transfer. Information on pathways, reactions and metabolites are gathered on databases, such as KEGG/LIGAND (Kanehisa and Goto, 2000; Goto et al., 2002) and BioCyc/EcoCyc (Karp et al., 2004). Currently, however, metabolic databases do not contain comprehensive information on atom correspondences across chemical reactions. KEGG RPAIR database (Kotera et al., 2004) contains such correspondences between single metabolite pairs. However, these are difficult to extend to mappings concerning the whole reaction without extensive manual work. In most other databases, atom mapping information is missing altogether.

[1]Department of Computer Science, University of Helsinki, Helsinki, Finland. [2]Nokia Research Center, Palo Alto, California.

Atom mappings of reactions have various uses and potential applications. Reconstruction of metabolic networks (Duarte et al., 2007; Pitkänen et al., 2008) is typically only done in metabolite level, whereas atom-level representation (ARM Arita, 2003) of the pathways would facilitate better understanding of metabolic networks. Reaction atom maps are a requisite to do consistency checking of pathways (Arita, 2005). Another application of the atom mappings is the global analysis of metabolic networks by computation of conservation ratios of atoms in metabolic reactions (Hogiri et al., 2008). Reactions can be classified based on their chemical transformations (Yamanishi et al., 2009). In drug design, predicting the fate of all parts and atoms of the candidate drug through the transformation pathways of the drug is useful for optimizing drug design. Tracer experiments, where a set of atoms is labeled chemically or isotopically, enable the tracing of atoms across reaction network (Arita, 2003; Menküc et al., 2008). In the ^{13}C flux analysis, an isotopically labeled substrate is fed to the cell and its pathways are traced by measuring the concentration of ^{13}C in different parts of the metabolic network. To be able to trace single atoms in the network, one must have knowledge of single atom's locations across the reactions (Rantanen et al., 2008, 2006; Rousu et al., 2003). Atom mapping information can be also used to deduce the relevant pathways a metabolite or drug is following (Blum and Kohlbacher, 2008).

The computational atom mapping problem has been tackled using mainly iterative maximum common substructure (MCS) approach, which has been extensively researched (Raymond and Willett, 2002). MCS is one of the well-established methods of graph matching (Bunke, 2000), which has been often applied towards chemical structures in the graph matching community (Sussenguth, 1965; Lynch, 1968; Levi, 1972; Cone et al., 1977; Tarjan, 1977; Lynch and Willett, 1978; McGregor and Willett, 1981; Xu, 1996; Wang and Zhou, 1997). The MCS approaches concentrate on finding largest intact substructures from the reactants and products of a reaction. These substructures are determined to undergo the reaction intact. Thus, a greedy series of maximal common substructure searches are made to deduce the corresponding atom regions of the reactants and products, producing the atom mapping.

Akutsu (2004) was the first to formalize the atom correspondences across chemical reactions and prove that the problem of atom mapping is NP-hard in general case. They concentrated on a special case where a single cut is made in the reaction, i.e., reactions are of type $X{-}A + Y{-}B \leftrightarrow X{-}B + Y{-}A$, where X, Y, A, and B are chemical species. They used Morgan's algorithm (Wipke and Dyott, 1974) to approximate the graph isomorphism algorithm on all 2-partitions of the reactant and product graphs. Arita used a modified Morgan's algorithm and an exhaustive search to find the approximated maximum common substructures without restrictions on the number of cuts (Arita, 2000). The ARM database of up to 3000 reactions was assembled with heavy curation containing mappings of carbon, phosphorus and nitrogen atoms only (Arita, 2003).

The SIMCOMP program (Hattori et al., 2003a) computes pairwise reactant-product mappings. They used maximum common substructure algorithm by first transforming the reactants and products into an association graph and by finding maximum cliques, which correspond to maximal common substructures in the original graphs, in the association graph (Hattori et al., 2003a,b). A modified Bron-Kerbosch maximal clique algorithm was used (Bron and Kerbosch, 1971; Koch, 2001; Cazals and Karande, 2005, 2008). SIMCOMP program was run against KEGG LIGAND database, and an RPAIR database of the pairwise mappings was generated for roughly 7000 reactant-product pairs (Kotera et al., 2004; Kanehisa et al., 2006). SIMCOMP doesn't take reaction information into account, and thus the pairwise mappings can't easily be extended to whole reaction mappings. Mu et al. (2007) used SIMCOMP to produce an updated set of atom mappings.

The MCS approach has several drawbacks. First, computing MCS is computationally hard, and the state-of-the-art methods either employ a cut-off limit (Hattori et al., 2003a) or approximated methods (Arita, 2003; Akutsu, 2004). Thus, it can be shown that MCS approach fails to find maximal subregions (Arita, 2003). Second, the MCS approach leaves the optimization criteria undefined and in practise heuristically approximates the minimum graph edit distance function.

Very recently, Crabtree and Mehta (2009) presented an atom-mapping method that is not based on subgraph isomorphism but on graph isomorphism. They formulated the atom-mapping problem as minimization of bonds broken and formed to realize the chemical reaction. They presented both exact and heuristic combinatorial algorithms to solve the problem. In this article, we adopt the same optimization-based philosophy. We present an efficient A* based algorithm (Hart et al., 1968; Dechter and Pearl, 1985) for automatic mapping of reaction atoms. No constraints on the reaction type or size is imposed. The algorithm's objective function is to find an atom mapping minimizing the graph edit distance. The algo-

Table 1. Atom and Bond Spectra of Cysteine with Hydrogens Omitted (Fig. 1)

			C	O	N	S		
$\alpha(G)$	=	[	3,	2,	1,	1	]	
			$C–C$	$C–O$	$C=O$	$C–N$	$C–S$	
$\beta(G)$	=	[	2,	1,	1,	1,	1	]

rithm directly maps individual atoms according to a set of constraints by a search procedure that is bounded by upper and lower bound functions of the edit distance. The search is guided by atom's topological context feature information, which was in a smaller and more restricted scale introduced by Hattori et al. (2003a). The objective function of the algorithm can be easily extended or altered to, for example, include information on thermodynamic energies of reactions (Mavrovouniotis, 1991; Blanskby and Ellison, 2003).

2. Atom Mapping Via Graph Edit Distance Minimization

In this section, we formally define the atom mapping problem that is to be solved by the algorithms presented in later sections.

A *molecule graph* $G=(V,E,L,W)$ is a (possibly disconnected) graph with nodes $i \in V$ labeled by chemical elements $L(i) \in \{H,C,N,O,\ldots\}$ and edges[1] $(i,j) \in E$ corresponding to covalent bonds, with integer weights $W(i,j)=0,1,2,3$ corresponding to the bond order (no bond, single, double, triple). Throughout this article, hydrogen atoms are omitted from the molecular graphs for simplicity.

The *atom spectrum* of G is the vector $\alpha(G)=(\alpha_l(G))_l$, where $\alpha_l(G)$ is the number of atoms $i \in V$ with label $L(i)=l$. The *bond type* of a pair of atoms with $W(i,j)>0$ is the triplet $T(i,j)=(L(i), L(j), W(i,j))$ denoting the adjacent atom labels and the bond order. A *bond spectrum* $\beta(G)=(\beta_t(G))_t$ is a vector where $\beta_t(G)$ denotes the number of bonds of type t in molecule graph G (Table 1, Fig. 1).

A *reaction* is a triplet $\rho=(R,P,M)$, where the R (resp. P) is a molecule graph called the *reactant graph* (resp. *product graph*), representing the *set of reactant* (resp. product) molecules. The *atom mapping* M for a pair (R,P) is a relation $M \subset V_R \times V_P$, where $(i,j) \in M$ denotes that a reactant atom i is mapped onto product atom j. The domain of M is denoted $dom(M)=\{v|(v,u) \in M\} \subset V_R$ and the range $ran(M)=\{u|(v,u) \in M\} \subset V_P$. An atom mapping is *complete* if $dom(M)=V_R$ and if $ran(M)=V_P$, that is, every reactant atom is mapped to at least one product atom and vice versa. An atom mapping is *valid* if the node labels of all mapped atoms agree: for all $(i,j) \in M, L_R(i)=L_P(j)$. A reaction $\rho=(R,P,M)$ is valid if M is valid and complete and the graphs R and P have equal atom spectra.

A *partial atom mapping* $M^* \subset M$ induces a partition of the both reactant and product graphs into mapped and residual subgraphs. The mapped reactant graph $R(M^*)$ consists of reactant edges $E_{R,M^*}=E_R \cap dom(M^*)^2$ for which both end points have been mapped, and the nodes induced by the edges. The residual reactant graph $\bar{R}(M^*)$ consists of remaining edges $E_R - E_{R,M^*}$ and the nodes induced by them. The set of nodes belonging to both the mapped and the residual graphs are called the *boundary*. The *complementary residual graph* $\bar{R}^C(M^*)$ consists of unmapped nodes $R(M^*) - \bar{R}(M^*)$ and the edges between them. The partition of the product graph into the mapped $P(M^*)$ and residual part $\bar{P}(M^*)$ is defined in analogous manner (Fig. 2).

In general, an atom mapping does not need to be bijective; i.e., a reactant atom can be mapped onto several product atoms (or vice versa), for example, when the participating molecules have symmetric orientations. Here, however, we restrict our attention to bijective atom mappings. For bijective mappings, we use the function/inverse function shorthands $M(i)=j$ and $M^{-1}(j)=i$ for $(i,j) \in M$. For example, see Table 1 for atom and bond spectra of cysteine.

The atom mapping problem is defined as follows.

[1] All edges are considered undirected; thus the order of stating the nodes of an edge does not matter, $(i,j) \equiv (j,i)$.

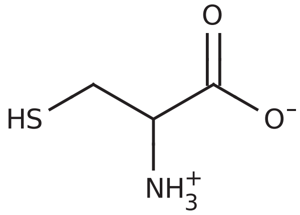


FIG. 1. Amino acid cysteine.

Definition 1. (Atom mapping problem). *Given a pair* (R, P) *of molecule graphs that have equal atom spectra, return all atom mappings* $M \subset V_R \times V_P$ *such that* (R, P, M) *is a valid reaction and the mapping cost is minimal, i.e.* $f(M) \leq f(M')$ *for any valid and complete atom mapping* $M' \subset V_R \times V_{P.}$

Ideally, the cost $f(M)$ assigned to an atom mapping M should correlate with the difficulty of making certain reaction to happen, or the probability of the reaction happening by chance. Accurate quantitative modelling of this kind is, however, computationally very demanding, as they require modelling of the energy landscapes of chemical reactions, which at the extreme case require quantum chemistry techniques (Gao et al., 2006). In systems biology applications, where thousands of enzymatic reactions need to be handled, such models are not tractable. Instead, simple heuristic cost functions needs to be used.

In this paper, we use a version of graph edit distance named *edge edit distance*:

Definition 2. (Edge edit distance). *Given a pair of graphs* G_1, G_2 *the edge edit distance* $d_{EE}(G_1, G_2)$ *is the minimum number of edge edit operations that is required to transform* G_1 *to* G_2.

An easy extension of edge edit distance is to assign weights to edges, so as to indicate which edges might be more difficult to cut than others.

Definition 3. (Weighted Edge Edit Distance). *Given a pair of graphs* G_1, G_2 *the weighted edge edit distance* $d_{WE}(G_1, G_2)$ *is the minimum sum of weights of inserted, deleted and renamed edges that is required to transform* G_1 *to* G_2.

The weights could, for example, correspond to bond order $W(i, j)$ or some other measure that correlates with the difficulty of editing the bond.

It should be clear that computing either edge edit distance variant is no easier than solving graph isomorphism: $d_{EE}(G_1, G_2) = 0$ if and only if G_1 and G_2 are isomorphic (resp. for d_{WE}). As there is no known polynomial algorithm for graph isomorphism (Uehara et al., 2005), we will not look for such for the edge edit distance in this article, but the focus is on general search algorithms that behave well in practice.

If the edit operations are restricted to insertions and deletions, given an optimal atom mapping M, the edge edit distance can be expressed as

$$f_{EE}(M) = d_{EE}(R, P) = |E_P \Delta E_M|,$$

where E_M is a set of images of the reactant graph edges under the mapping M, that is, $e = (M(i), M(j))$ for some $(i, j) \in E_R$. Similarly, the weighted edit distance is given by

$$f_{WE}(M) = d_{WE}(R, P) = \sum_{e \in E_P \Delta E_M} W(e).$$

Here, the edges in the set E_M inherit their weights from the reactant graph $W(M(i), M(j)) = W(i, j)$.

3. ATOM MAPPING ALGORITHM

In this section, we will describe a fast algorithm for computing atom mappings minimizing the edit distance based costs $f_{EE}(M)$ and $f_{WE}(M)$.

Our algorithm is made of the following ingredients, which we will detail subsequently:

- An A* type total path cost estimate to guide the search in the space of partial atom mappings.
- An extension operator for partial mappings that maintains the path cost estimates in constant time per edge.
- Pruning of A* search space by computing upper bounds on the optimal cost via fast greedy search.

3.1. Total path cost estimates for partial atom mappings

We solve the atom mapping problem via search in the space of partial mappings. Thus, the states correspond to valid partial mappings $M^* \subset M$, where some atoms are already mapped and the rest are not. A state transition corresponds to augmenting a partial mapping by mapping one unmapped reactant atom to an unmapped product atom.

Here we derive an A* type path cost estimate for this partial mapping space. The estimate is divided into two components

$$\hat{f}(M^*) = g(M^*) + h(M^*),$$

the *accumulated cost* $g(M^*)$ so far is the cost to arrive at the partial mapping M^*, and the *future cost* estimate, a lower bound for the cost that will still be accumulated to complete the mapping. The consequence of $h(M^*)$ being a lower bound on the future cost is that $\hat{f}(M^*)$ is a lower bound for the total path cost $f(M)$:

Lemma 1. *If $M^* \subset M$ then $f(M) \geq g(M^*) + h(M^*) = \hat{f}(M^*)$*

In this article, the accumulated cost will be given as the edge edit distance of the mapped reactant and product subgraphs:

$$g(M^*) = f_{EE}(R(M^*), P(M^*))$$

For the future cost estimate, two properties are essential: First, it should be tight enough to prune suboptimal branches from the search space. Second, it should be fast to evaluate so that benefits of search space pruning will realize. We will give three alternatives for future cost estimate in the following. For bijective atom mappings, it turns out that practical future cost estimates can be obtained by using the information contained either in the bond spectra or in the atom spectra of the reactant and product graphs.

For the first estimate, the bond spectrum difference

$$\Delta\beta(R, P) = \beta(R) - \beta(P)$$

of the reactant and product graphs is the key concept. We set the future cost estimate

$$h_\beta(M) = \sum_t |\Delta\beta_t(\bar{R}(M), \bar{P}(M))|$$

to be the bond spectrum difference of the residual graphs of M. The following lemma established the lower bound property

Lemma 2. *If M is a valid bijective partial atom mapping, then*

$$f_{EE}(\bar{R}(M), \bar{P}(M)) \geq h_\beta(M)$$

Proof. The result follows from the simple observation that one edit operation can change the bond spectrum difference by at most one, and that both sides of the inequality are non-negative by definition. ■

Any possible looseness in the above bound is caused by situations where some extra edit operations that do not decrease bond spectrum difference are needed to make the mapping topologically feasible. We note

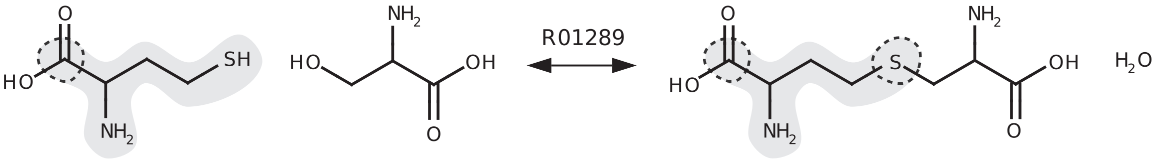


FIG. 2. KEGG Reaction R01289: `serine+homocysteine <=> cystathione+H2O.` Gray regions show the partial mapping M^*. Unhighlighted areas together with dashed nodes indicate the residual reactant and the residual product graphs. The unhighlighted areas only form the complementary residual graphs.

that this bound can be evaluated efficiently: for each partial mapping, we keep track of the bond spectra of the residual graphs in a hash table and their current bond spectrum difference. Both can be updated in near constant time when expanding the partial mapping.

Another lower bound can be obtained by examining the neighborhoods of atoms in the reactant and product graphs. Let $\gamma(G)$ denote the *neighborhood spectrum* of molecule graph G, defined by $\gamma(G)=(\gamma_t(G))$, where $\gamma_t(G)$ is the count of atom neighborhoods of type t in G. Atom neighborhood type is defined as a pair $t=(l,s)$, where l is an atom type and $s\in L^{k(l)}$ is a string of alphabetically ordered atom type labels and $k(l)$ is the maximum number of neighbors for atom type l. For example, $t=(C,\ COO)$ describes a carbon connected to one carbon and two oxygen atoms, and $\gamma_{(C,\ COO)}(G)$ represented the number of such arrangements in G. The future cost estimate

$$h_\gamma(M^*)=\frac{1}{2}\sum_t|\Delta\gamma_t(\bar{R}^C(M^*),\bar{P}^C(M^*))|$$

is given as the sum of absolute differences in the neighborhood spectra of the complementary residual graphs, that is, in the unmapped regions of reactant and product graphs. The lower bound property is given by

Lemma 3. *If M is a valid bijective partial atom mapping, then*

$$f_{EE}(\bar{R}(M),\bar{P}(M))\geq h_\gamma(M)$$

Proof. The lemma follows from the observation that one edge edit operation changes exactly two atom neighborhoods and hence can change the neighborhood difference by at most two, and from the non-negativity of both sides of the inequality. ■

The future cost estimate h_γ is similarly efficient to evaluate: we can incrementally update the neighborhood spectra of the residual graphs and the current neighborhood spectrum difference in near constant time.

The two lower-bound can be combined to a single bound by taking the maximum. Thus, the final future cost estimate used in our algorithm is given by

$$h(M^*)=\max\{h_\beta(M^*),h_\gamma(M^*)\}.$$

Example. The KEGG reaction R01289 `serine+homocysteine <=> cystathione+H2O` is part of the cysteine synthesis pathway (Fig 2). A partial mapping M^* is highlighted in the figure. The partial mapping's accumulated cost is $g(M^*)=0$ as the partially mapped regions of reactant and product sides are equal. The future cost estimate based on bond spectrum is $h_\beta(M^*)=2$, as the bond spectrum's of residual graphs differ by an extra C—S and a missing C—O on the reactant side (Table 2). The future cost estimate based on atom neighborhoods is $h_\gamma(M^*)=2$ (Table 3). The cost is thus $\hat{f}(M^*)=g(M^*)+h(M^*)=0+2=2$.

Finally, we note that a lower bound can be derived by performing a minimum weight bipartite matching between the reactant and product atoms in the residual graph of a partial mapping. We construct a bipartite graph $G=(\bar{V}_R(M),\bar{V}_P(M),E,W)$ taking the nodes of the residual reactant and product graphs and connecting nodes with edges $E=\{(r,p)|r\in\bar{V}_R(M),p\in\bar{V}_P(M),L(r)=L(p)\}$, edge weights given by W.

Table 2. Bond Spectra of the Mapped Graph, Residual Graph, Whole Graph, and Their Difference

			$C-C$	$C-O$	$C=O$	$C-N$	$C-S$	
$\beta(\bar{R}(M^{*}))$	$=$	[	2,	3,	2,	1,	0	]
$\beta(R(M^{*}))$	$=$	[	3,	0,	0,	1,	1	]
$\beta(R)$	$=$	[	5,	3,	2,	2,	1	]
			$C-C$	$C-O$	$C=O$	$C-N$	$C-S$	
$\beta(\bar{P}(M^{*}))$	$=$	[	2,	2,	2,	1,	1	]
$\beta(P(M^{*}))$	$=$	[	3,	0,	0,	1,	1	]
$\beta(P)$	$=$	[	5,	2,	2,	2,	2	]
$\Delta\beta(\bar{R}(M^{*}),\bar{P}(M^{*}))$	$=$	[	0,	+1,	0,	0,	-1	]
$h_{\beta}(M^{*})$	$=$	2						

Any bipartite matching, given by a subset $B\subset E$ of edges that define a one-to-one in $\bar{V}_R(M)$, directly gives a valid atom mapping. However, in general it does not need to be the one with minimum edge edit distance. The suboptimality arises from the fact that the edges are matched independently, disregarding the fact that mapping a pair of adjacent reactant atoms to a pair of non-adjacent product atoms necessarily induces at least one edge edit.

However, by setting the bipartite weights appropriately, we can guarantee the bipartite matching cost bound the edit distance from below. This is achieved as follows.

For each edge (r,p) we examine the bond spectrum differences in the neighborhood graphs $G_{\mathcal{N}}(r)$ and $G_{\mathcal{N}}(p)$ induced by the bonds adjacent to r, and p, respectively. We keep separately track of the positive and negative differences, $\Delta\beta_{\mathcal{N}}(r,p)_{+}=\sum_t \max(0,\beta_t(G_{\mathcal{N}}(r))-\beta_t(G_{\mathcal{N}}(p)))$ and $\Delta\beta_{\mathcal{N}}(r,p)_{-}=\sum_t \max(0,\beta_t(G_{\mathcal{N}}(r))-\beta_t(G_{\mathcal{N}}(p)))$. The sum is set as the edge weight in the bipartite graph:

$$W(r,p)=\frac{1}{2}(\Delta\beta_{\mathcal{N}}(r,p)_{+}+\Delta\beta_{\mathcal{N}}(r,p)_{-}).$$

The coefficient 1/2 comes from the fact that neighborhood differences are divided equally among the two end points of a bond. The bipartite matching cost is then

$$h_{BPM}(M)=\sum_{(r,p)\in B} W(r,p).$$

We have the following lemma:

Lemma 4. *If M is a valid bijective atom mapping, then*

$$f_{EE}(\bar{R}(M),\bar{P}(M))\geq h_{BPM}(M) \tag{1}$$

Proof. We can equivalently write $f_{EE}(R,P)=\sum_{r\in R}\sum_{t\in T} f^{t}_{EE}(r)$, where $f^{t}_{EE}(r)=\frac{1}{2}|\{r'\in R|(r,r')\in EE, L(r')=t\}|$ is the number of edit operations in the bonds adjacent to r divided by two, and $f^{t}_{BPM}(r,p)=\frac{1}{2}|\Delta\gamma_t(r,p)|$ is the contribution of the atom type t to the cost of the edge (r,p) in the bipartite mapping.

Assume now contrary to the claim that $f_{EE}(R,P)<f_{BPM}(R,P)$. Then we can find a such atom r and atom type t that $f^{t}_{EE}(r)<f^{t}_{BPM}(r,p), f^{t}_{BPM}(r,p)>0$, and $(r,p)\in M$. Thus there is a surplus of $k=f^{t}_{FBM}(r,p)$

Table 3. The Atom Neighborhoods of Residual Graphs of Partial Mapping M^{*} Indicated in Figure 2

			(C,CO)	(C,CS)	(C,CCN)	(C,COO)	(O,C)	(O,)	(N,C)	
$\gamma(\bar{R}^{C})$	$=$	[	1,	0,	1,	1,	5,	0	1	]
$\gamma(\bar{P}^{C})$	$=$	[	0,	1,	1,	1,	4,	1,	1	]
$\Delta\gamma(\bar{R}^{C},\bar{P}^{C})$	$=$	[	+1,	-1,	0,	0,	+1,	-1,	0	]
$h_{\gamma}(M^{*})$	$=$	2								

The atom neighborhoods differ at four locations, and thus we need at least two bond changes to equalize them.

neighbors of type t in the reactant neighborhood as compared to the product neighborhood. As M is a valid mapping, k atoms of type t cannot be mapped to the neighbors of p, instead the bonds should be cleaved by edit operations. However, by our assumptions in the neighborhood of r there are only $f^t_{EE}(r) < k$ edit operations. Hence, after using all edit operations, there will be $f_{BPM}(r,p) - f_E E^t(r)$ neighbors remaining that have not been cleaved and cannot be mapped to neighbors of p. This is a contradiction and hence our claim is proven. ■

Although (1) gives a relatively tight lower bound for the future cost, it comes with a price: computing the bipartite matching takes $O(V^3)$ time (Munkres, 1957; Riesen et al., 2007). Hence, it is believed to be too slow to be used within the A* algorithm.

3.2. *Efficient expansion of partial mappings*

Given an arbitrary partial mapping M, how do we efficiently choose the next pair of atoms to be mapped? There are two issues to consider:

- We do not want to revisit a state that we have already visited. Blindly letting any unmapped atom to be a candidate would make us, at the worst case, visit each partial mapping M as many times as there are different permutations.
- We wish to maintain the total path cost estimate efficiently. This means that the effect of the newly mapped atom to both the accumulated cost and the future cost estimate should be fast to compute.

In our approach the above is achieved by numbering the reactant atoms consecutively using breadth-first search. We start from an extreme atom of the largest reactant and iteratively process the rest of the reactants in the order of their size. On the product side, no ordering for the atoms is imposed but any correctly labeled unmapped atom can be paired with the next reactant atom. Thus, the search tree only branches on the choice of the product atoms, not on the choice of reactant atoms (see Fig. 6 below).

This approach ensures that the same states are not revisited. The efficient maintenance of the total path cost estimate is as follows: when the atom pair (r,p) is added to the partial mapping M, the incurred edge edit distance is given by the sum of mapped neighbors r' of r whose images $M(r')$ are not neighbors of r and the mapped neighbors p' of p that are not neighbors of r, that is the symmetric difference of the neighbor sets. Computation of this number entails single traversal of the neighbor sets of the newly mapped atoms that takes constant asymptotic time as the neighbor sets have size at most four due to chemical valence rules.

3.3. *Pruning the search*

The A* algorithm (Algorithm 1) despite its theoretical appeal, has one practical weakness: the priority queue of partial solutions can grow very large and the available main memory will become a limiting factor on how complex reactions can be mapped.

Fortunately, it is possible to prune the priority queue by a simple strategy: we can use a fast heuristic algorithm to complete any partial solution. This procedure will give us a complete, valid atom mapping M with some cost $f(M)$. The obtained cost $f(M)$ is obviously an upper bound for the optimal cost and all partial solutions whose lower bound is already higher than the upper bound can be pruned from the priority queue.

Here, we present a pruning strategy based on two heuristic algorithms, greedy search and bipartite matching. The greedy algorithm augments the partial mapping by always mapping the pair of atoms with least increase in total path cost (Algorithm 2). The bipartite matching algorithm, on the other hand, matches the atoms independently, guided by a similarity score for atoms.

3.3.1. Atom features. Both our heuristic mapping algorithms benefit from a similarity score s for atoms: in the greedy algorithm, atom similarity guides the selection of which atom pair to map the next. In the bipartite matching algorithm, the similarity score is used in defining the weight of the individual atom matchings. The simplest similarity score is to only allow matching of atoms of equal label L. In practice, it is useful to also compare the neighborhoods of atoms i and j to be matched. The neighborhoods differ if the atoms lie at the chemical reaction's site-of-modification (SOM), but usually a major part of the molecule is not immediately touched by the chemical transformations. In these areas, the neighborhoods of matched atoms are equivalent up to the distance to the closest SOM.

Algorithm 1 $S \leftarrow$ ASTARMAP (R, P)

```
Inputs: Reactant graph $R$, product graph $P$
Outputs: Set $S$ of optimal atom mappings
  $(r_1, r_2, \ldots, r_k) \leftarrow$ Order the reactant atoms using BFS
  $S \leftarrow \emptyset$; $ub \leftarrow \infty$
  Priority queue $Q \leftarrow \{(r_1, p) | L(r_1) = L(p),\ p \in P\}$
  while $Q$ is not empty do
    $M \leftarrow removeBest[Q]$ {minimizing $\hat{f}(M) = g(M) + h(M)$}
    if $\hat{f}(M) > ub$ then
      return $S$ {only worse solutions left in $Q$, we can finish}
    end if
    if $M$ is complete then
      $ub \leftarrow f(M)$
      $S \leftarrow S \cup \{M\}$ {this is an optimal mapping}
    else
      $M' \leftarrow expandMapping(M)$
      $insertQueue(Q, M')$;
    end if
end while
return $S$
```

An similarity score based on atom's k-neighborhoods is presented. It is composed of four atom features:

1. Atom distribution $ad_k(i)$ defines the distribution of atoms up to distance k around the atom i. Particularly $ad_0(i) = L(i)$ identifies the label of the atom itself.
2. Wiener index (Bereg, 2008) $wiener_k(i)$ is the sum of distances between all vertex pairs in a graph spanned by distance k from atom i. Wiener index is an invariant related to the branching properties of the graph.
3. Morgan index (Bereg, 2008; Wipke and Dyott, 1974) $morgan_k(i)$ is an iteratively computed topological invariant of the molecule graph computed for a submolecule spanned by distance k from atom i.
4. Finally, the fourth atom feature $ring_k(i)$ tells whether atom i is part of a k-sized ring for $k \in \{4, 5, 6\}$.

The similarity score indicates how far from the i and j the neighborhoods are equivalent according to different atom features. Each feature has an equal amount of weight in the similarity score s_k, where k is the size of the neighborhood:

Algorithm 2 *GreedyMap* (R, P)

```
  $M \leftarrow$ empty map
  $reacAtoms \leftarrow V_R$
  $prodAtoms \leftarrow V_P$
  while $V_R \setminus dom(M) \neq \emptyset$ do
    $minV al \leftarrow \infty$
    $minPair \leftarrow NIL$
    for all $(r, p) \in reacAtoms \times prodAtoms$ do
      if $f_c(r, p) < minVal$ then
        $minVal \leftarrow f_c(r, p)$
        $minPair \leftarrow (r, p)$
      end if
    end for
    $M \leftarrow M \cup minPair$
    remove p from $reacAtoms$
    remove r from $prodAtoms$
  end while
```

$$\begin{aligned} s_k(i,j) = {} & \frac{1}{4}\sum_{l=1}^{k}[ad_l(i) = ad_l(j)] \\ & + \frac{1}{4}\sum_{l=1}^{k}[wiener_l(i) = wiener_l(j)] \\ & + \frac{1}{4}\sum_{l=1}^{k}[morgan_l(i) = morgan_l(j)] \\ & + \frac{1}{4}\sum_{l=4}^{6}[ring_l(i) = ring_l(j)]. \end{aligned}$$

Above, [·] denotes the indicator function. The value of k was set to 10 to differentiate the large intact regions of large molecules. We have precomputed the atom features for the whole KEGG LIGAND database and thus the computation of similarity score is made in constant time during the A* search.

The similarity function is used in greedy algorithm to differentiate between all candidate atom-pairs to be added to the partial mapping, if there are several atom-pairs which have the minimal effect on the cost estimate $\hat{f}(M^*)$. In bipartite graph matching, the similarity function serves as the edge weight.

In the A* algorithm, similarity function is also used to differentiate between partial mappings with equally lowest estimated score $\hat{f}(M^*)$. Each partial mapping in the A* queue represents an addition of atom pair to the previous partial mapping and the similarity function tells which of the candidate atom pairs are most similar with respect to their extended neighborhoods (see Fig. 6 below).

We note that the MCS approach by Hattori et al. (2003a, b) also uses a similarity function by labeling the atoms into 68 groups based on their immediate or ring surroundings.

4. EXPERIMENTS

The experiments were done on KEGG/Ligand database version 49 (January 9, 2008) containing a set of 7781 common biotransformation reactions. A total of 6015 reactions contained full and complete definition of the biotransformation. Rest of the reactions had missing, erroneous, ambiguous or unbalanced reaction definitions. These valid reactions were mapped using four algorithms: A*, maximum common subgraph (MCS) algorithm used by Hattori et al. (2003a) (with iteration cutoff parameter $R = 100{,}000$), Bi-partite graph matching algorithm (implementation by Hungarian algorithm [Munkres, 1957; Riesen et al., 2007]) and naive greedy algorithm. All algorithms were implemented using Java and computed with 4 Gb of memory and Intel Xeon X5355 cpu running at 2.66 GHz. The A* algorithm managed to compute 5802 reactions, and our MCS implementation 5934 reactions, of the 6015 valid KEGG reactions in less than one hour per reaction. BPM and greedy algorithms computed all reactions. A total of 5624 reactions were computed with all algorithms and are comparable.

As the A* algorithm finds an optimal atom mapping with respect to the edit distance, the main result is to compare how often MCS-based approach errs from this on real biochemical reactions represented by KEGG. We also analyze the performance of greedy and BPM procedures, and RPAIR pairwise mappings with respect to their edit distances.

Figure 3 shows the count of reactions with respect to their difference from optimal edit distance. MCS algorithm optimally maps 5119 (91.0%) reactions. The difference is +2 for 359 reactions (6.4%), +4 for 66 (1.2%) and larger for 80 (1.4%) reactions.

The two procedures suggested as pruning algorithms are of comparable accuracy. BPM and greedy algorithms achieve 63.3% and 54.4% optimal accuracy, respectively. They differ from MCS by having a longer and wider tail. Both procedures are highly dependent on the performance of the similarity function and atom features. Both algorithms work well as pruning procedures where they are used numerous times during the A* search.

Figure 4 shows the performance of the four algorithms. The reaction mappings are sorted by edit distance individually for each algorithm. The greedy and BPM algorithm's results rise fast, while the MCS algorithm is effective for the majority of the reactions. The MCS also has a high tail indicating mappings where

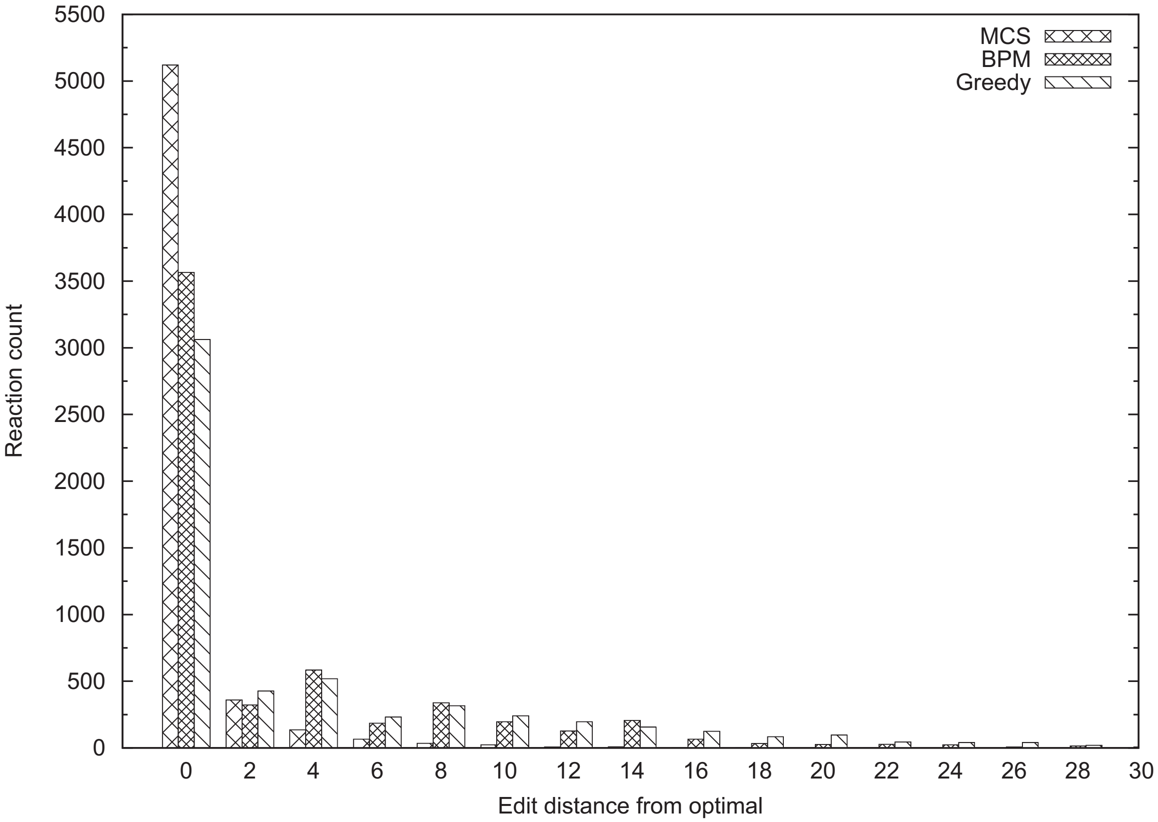


FIG. 3. Count of reactions with edit distance difference from optimal.

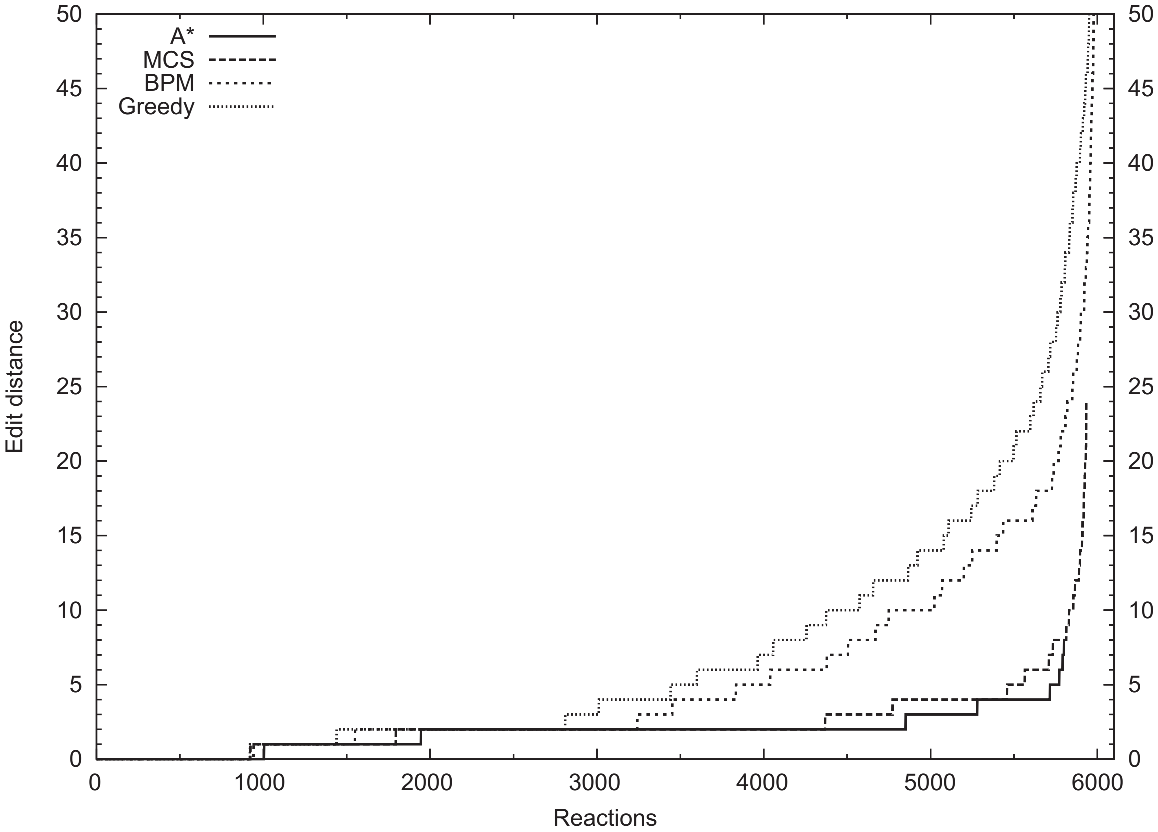


FIG. 4. Performance distributions of the four algorithms.

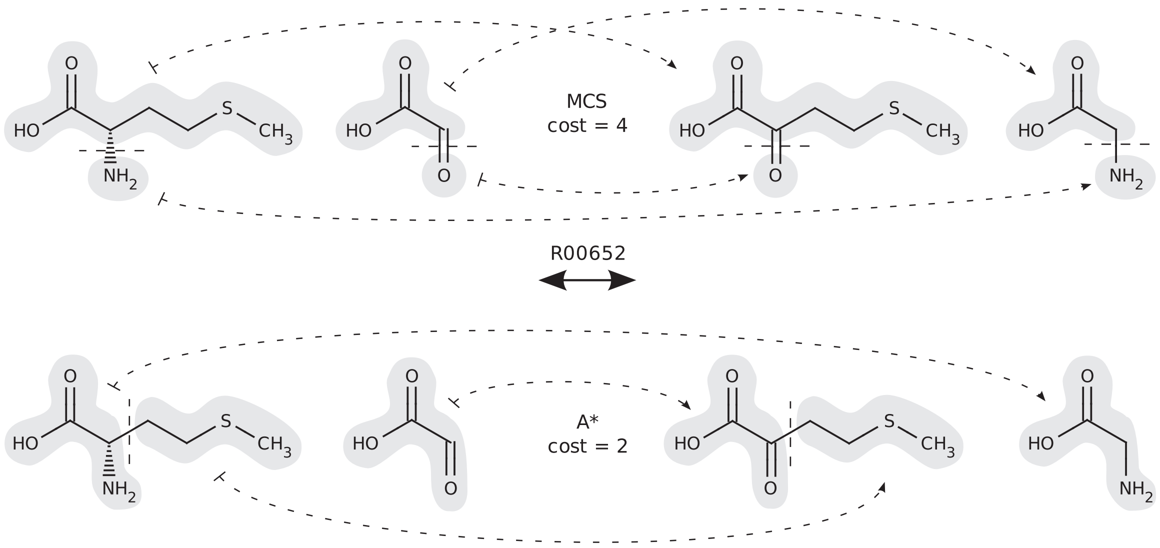


FIG. 5. Competing atom mappings of methionine:glyoxylate aminotransferase reaction. The MCS algorithm results in mapping of cost 4, while A* achieves a mapping of cost 2. However, the MCS mapping exhibits the correct reaction mechanism according to biochemical knowledge.

MCS procedure gives results of high edit distance. The MCS fails especially on reactions with high minimum edit distance. These reactions are often large and have complex reaction mechanism.

The running times of A* and MCS algorithms are comparable. Both algorithms suffer from high memory requirements, but in the A* algorithm smart pruning strategies can alleviate the problem. Also, because A* maintains both lower and upper bounds on the optimal solutions, insight into the optimality of the result achieved after early termination is acquired.

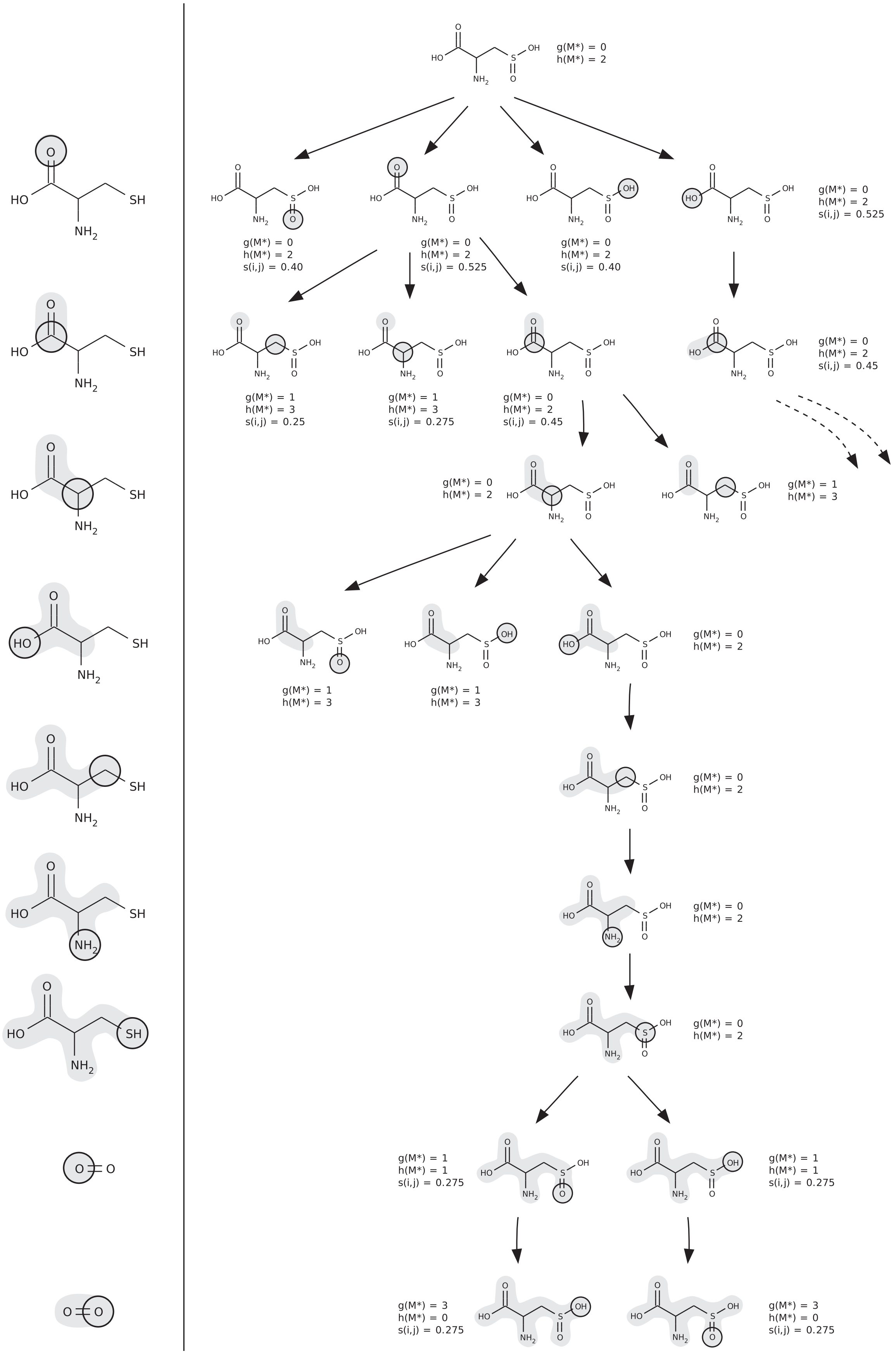
KEGG RPAIR database contains a total of 10124 reactant-product mappings. We reconstructed reaction level mappings by combining the pairwise mappings of all reactant-product pairs of each reaction. Most of the reconstructed mappings only partially cover the reactants and products, or are not bijective. A total of 6851 reactions mappings can be at least partially reconstructed and 4364 of those are complete mappings. However, RPAIR entries use internal molecular definitions which differ from the standard definitions used in KEGG LIGAND. We couldn't automatically match these two sources of molecular definitions, and hence only 2161 of the reconstructed mappings are consistent with KEGG LIGAND. These mappings are generated with SIMCOMP MCS algorithm and are manually curated (Kotera et al., 2004).

1932 of these mappings have the optimal edit distance, while only 16 are non-optimal (all differing by 2). This is due to the manual curation of KEGG RPAIR. When these 2161 mappings are compared against the direct MCS mappings, 13 % of the reconstructed mappings have different edit distance compared to direct MCS mappings, signifying manual curation. 3% of the curated mappings have larger edit distance than uncurated mappings, while 97% have smaller edit distance. This indicates that domain experts almost always prefer the mappings with low edit distances.

5. DISCUSSION

Little is known about the process of enzymatic reactions with respect to the resulting atom mapping. In general, the enzyme doesn't need to minimize the edit distance—or to maximize the size of common

FIG. 6. Search tree of the A* algorithm on R00893: cysteine+oxygen <=> 3-sulfino-alanine. The left column indicates with circles the reactant graph atom ordering, which is fixed. The tree shows possible partial mappings at each step starting from the empty mapping and progression of the A*. The algorithm chooses always the partial mapping which minimizes the cost function and find two optimal mappings fast. On the first row similarity function is used to distinguish better candidate. After the two complete mappings are found, the algorithm continues from the root to another candidate with $f = 2$ to ensure that all remaining optimal mappings are found.


g(M*) = 0
h(M*) = 2
g(M*) = 0
h(M*) = 2
s(i,j) = 0.40
g(M*) = 0
h(M*) = 2
s(i,j) = 0.525
g(M*) = 0
h(M*) = 2
s(i,j) = 0.40
g(M*) = 0
h(M*) = 2
s(i,j) = 0.525
g(M*) = 1
h(M*) = 3
s(i,j) = 0.25
g(M*) = 1
h(M*) = 3
s(i,j) = 0.275
g(M*) = 0
h(M*) = 2
s(i,j) = 0.45
g(M*) = 0
h(M*) = 2
s(i,j) = 0.45
g(M*) = 0
h(M*) = 2
g(M*) = 1
h(M*) = 3
g(M*) = 1
h(M*) = 3
g(M*) = 1
h(M*) = 3
g(M*) = 0
h(M*) = 2
g(M*) = 0
h(M*) = 2
g(M*) = 0
h(M*) = 2
g(M*) = 0
h(M*) = 2
g(M*) = 1
h(M*) = 1
s(i,j) = 0.275
g(M*) = 1
h(M*) = 1
s(i,j) = 0.275
g(M*) = 3
h(M*) = 0
s(i,j) = 0.275
g(M*) = 3
h(M*) = 0
s(i,j) = 0.275

subgraphs. The reaction mechanism is the result of chemical and physical interactions and laws, which in turn produce the atom mapping pattern. An interesting approximation of more realistic cost function is to use bond dissociation energies (Blanskby and Ellison, 2003), approximated with, for example, group contribution theory (Mavrovouniotis, 1991).

An example of a reaction which does more work than needed is R00652 `methionine:glyoxylate aminotransferase` (Fig. 5). Here, the real transformation is thought to happen as shown on top (Glover et al., 1988), which requires a total of four operations: two cuts and two new bond formations. This is the mapping the MCS algorithm finds. The A* algorithm finds the mapping shown on bottom, which includes only two operations. Here, the simpler mapping is unrealistic probably because of the high cost of cutting the methionine at the middle. In the MCS mapping, only easily modifiable oxygen and nitrogen atoms are operated at the edges of the molecules. The A* cost function could be extended to take this into account.

Enzymes catalyzing reactions can be classified into groups such as *transferases* or *isomerases* according to, for example, enzyme classification (EC) system. This contextual information could be used to infer the type of reaction and the likely type of transformation of the reaction. The structural information of the enzymes and substrates of a reaction could be used for 3D modeling and prediction of the likely site-of-modification. 3D modeling has been actively researched in drug design (de Graaf et al., 2005).

An interesting development to the framework presented here would be to explicitly model the *symmetry classes* of optimal atom mappings. That is, all mappings which are automorphic. As seen in Figure 6, symmetrical atoms lead to duplication of the number of optimal mappings. Handling of automorphic mappings can be either designed into an mapping algorithm or be done afterwards to classify the resulting mappings into symmetry classes. We have implemented an VF2-based isomorphism algorithm which can answer whether two bi-graph mappings (i.e., atom mappings) are isomorphic (Cordella et al., 1999, 2004). To our knowledge, no current method deals with symmetries directly in the mapping algorithm itself.

In a setting where only one optimal mapping is sufficient, the upper bound can be optimized. Instead of setting the upper bound to the cost of current best mapping, it can be set to $f(M) - 1$ as we are only interested in mappings which are better than the current best mapping. This leads to more effective pruning of the search space. This strategy is especially effective as the cost function f is discrete and often small (Fig. 4).

6. CONCLUSION

This article describes a novel A* algorithm for atom mapping problem. The proposed method isn't based on MCS, but utilizes a sophisticated cost function to determine the atom mapping that incurs the minimum number of edge operations. The algorithm uses smart heuristics to guide itself through the space of atom mappings and guarantees an optimal result. The A* algorithm is well-defined and can be modified to use any cost function to determine a good atom mapping. We also propose a novel set of atom features, which are used to distinguish between similar atoms. The algorithm was used against KEGG database, and the resulting mappings agree well with the RPAIR database mappings, which are manually curated.

AUTHOR CONTRIBUTIONS

M.H., T.M., and J.R. designed the algorithms. S.L. and M.H. implemented the algorithms. M.H. conducted the experiments. J.R. and M.H. designed the experiments and co-wrote the manuscript.

ACKNOWLEDGMENTS

This work was supported by the Academy of Finland (grant 118653; ALGODAN) and in part by the IST Programme of the European Community under PASCAL2 Network of Excellence (IST grant-2007-216886). This publication only reflects the authors' views.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Akutsu, T. 2004. Efficient extraction of mapping rules of atoms from enzymatic reaction data. *J. Comput. Biol.* 11, 449–462.

Arita, M. 2000. Graph modeling of metabolism. *J. Jpn. Soc. Artif. Intell.* 15, 703–710.

Arita, M. 2003. *In silico* atomic tracing by substrate-product relationships in *Escherichia coli* intermediary metabolism. *Genome Res.* 13, 2455–2466.

Arita, M. 2005. Introduction to the arm database: database on chemical transformation in metabolism for tracing pathways, 193–210. *In*: *Metabolomics, The Frontier of Systems Biology*. *Volume 4.* Springer, Tokyo.

Bereg, S. 2008. Topological indices in combinatorial chemistry, 419–463. *In*: *Bioinformatics Algorithms: Techniques and Applications*. John Wiley & Sons, Inc., New York.

Blanskby, S., and Ellison, G. 2003. Bond dissociation energies of organic molecules. *Acc. Chem. Res*. 36, 255–263.

Blum, T., and Kohlbacher, O. 2008. Using atom mapping rules for an improved detection of relevant routes in weighted metabolic networks. *J. Comput. Biol.* 15, 565–576.

Bron, C., and Kerbosch, J. 1971. Finding all cliques of an undirected graph. *Commun. ACM* 16, 575–577.

Bunke, H. 2000. Graph matching: theoretical foundations, algorithms, and applications. *Proc. Vision Interface* 82–88.

Cazals, F., and Karande, C. 2005. An algorithm for reporting maximal c-cliques. *Theor. Comput. Sci.* 349, 484–490.

Cazals, F., and Karande, C. 2008. A note on the problem of reporting maximal cliques. *Theor. Comput. Sci.* 407, 564–568.

Cone, M., Venkataraghaven, R., and McLafferty, F. 1977. Molecular structure comparison program for the identification of maximal common substructures. *J. Am. Chem. Soc.* 99, 7668–7671.

Cordella, L., Foggia, P., Sansone, C., et al. 2004. A (sub)graph isomoprhism algorithm for matching large graphs. *IEEE Trans. Patt. Anal. Mach. Intell.* 26, 1367–1372.

Cordella, L.P., Foggia, P., Sansone, C., et al. 1999. Performance evaluation of the vf graph matching algorithm. *Proc. ICIAP '99* 1172.

Crabtree, J.D., and Mehta, D.P. 2009. Automatic reaction mapping. *ACM J. Exp. Algorithms* 13.

de Graaf, C., Vermeulen, N.P.E., and Feenstra, K.A. 2005. Cytochrome p450 in silico: an integrative modeling approach. *J. Med. Chem.* 48, 2725–2755.

Dechter, R., and Pearl, J. 1985. Generalized best-first search strategies and the optimality of A*. *J. ACM* 32, 505–536.

Duarte, N.C., Becker, S.A., Jamshidi, N., et al. 2007. Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proc. Nat. Acad. Sci. USA* 104.

Gao, J., Ma, S., Major, D.T., et al. 2006. Mechanisms and free energies of enzymatic reactions. *Chem. Rev.* 106, 3188–3209.

Glover, J.R., Chapple, C.C.S., Rothwell, S., et al. 1988. Allylglucosinolate biosynthesis in *Brassica carinata. Phytochemistry* 27, 1345–1348.

Goto, S., Okuno, Y., Hattori, M., et al. 2002. LIGAND: a database of chemical compounds and reactions in biological pathways. *Nucleic Acids Res.* 30, 402–404.

Hart, P.E., Nilsson, N.J., and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybernet.* 4, 100–107.

Hattori, M., Okuno, Y., Goto, S., et al. 2003a. Heuristics for chemical compound matching. *Genome Inform.* 14, 144–153.

Hattori, M., Okuno, Y., Goto, S., et al. 2003b. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *J. Am. Chem. Soc.* 125, 11853–11865.

Hogiri, T., Furusawa, C., Shinfuku, Y., et al. 2008. Analysis of metabolic network based on conservation of molecular structure. *BioSystems* 95, 175–178.

Kanehisa, M., and Goto, S. 2000. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* 28, 27–30.

Kanehisa, M., Goto, S., Hattori, M., et al. 2006. From genomics to chemical genomics: new developments in kegg. *Nucleic Acids Res.* 34, 354–357.

Karp, P.D., Arnaud, M., Collado-Vides, J., et al. 2004. The *E. coli* EcoCyc Database: no longer just a metabolic pathway database. *ASM News* 70, 25–30.

Kell, D. 2004. Metabolomics and systems biology: making sense of the soup. *Curr. Opin. Microbiol.* 7, 296–307.

Kitano, H. 2002. Systems biology: a brief overview. *Science* 295, 1662–1664.

Koch, I. 2001. Enumerating all connected maximal common subgraphs in two graphs. *Theor. Comput. Sci.* 250, 1–30.

Kotera, M., Hattori, M., Oh, M.A., et al. 2004. Rpair: a reactant-pair database representing chemical changes in enzymatic reactions. *Genome Inform.* 15, P062.

Levi, G. 1972. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo* 9, 1–12.

Lynch, M. 1968. Storage and retrieval of information on chemical structures by computer. *Endeavour* 27, 68–73.

Lynch, M., and Willett, P. 1978. The automatic detection of chemical reaction sites. *J. Chem. Inform. Comput. Sci.* 18.

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

McGregor, J., and Willett, P. 1981. Use of a maximal common subgraph algorithm in the automatic identification of the ostensible bond changes occurring in chemical reactions. *J. Chem. Inform. Comput. Sci.* 21.

Menküc, B., Gille, C., and Holzhütter, H. 2008. Computer aided optimization of carbon atom labeling for tracer experiments. *Genome Inform.* 20, 270–276.

Mu, F., Williams, R., Unkefer, C., et al. 2007. Carbon-fate maps for metabolic reactions. *Bioinformatics* 23, 3193–3199.

Munkres, J. 1957. Algorithms for the assignment and transporation problems. *J. Soc. Indust. Appl. Math.* 5, 32–38.

Pitkänen, E., Rantanen, A., Rousu, J., et al. 2008. A computational method for reconstructing gapless metabolic networks, 288–302. *In*: *Bioinformatics Research and Development. Communications in Computer and Information Science.* Springer, New York.

Rantanen, A., Mielikäinen, T., Rousu, J., et al. 2006. Planning optimal measurements of isotopomer distributions for estimation of metabolic fluxes. *Bioinformatics* 22, 1198–1206.

Rantanen, A., Rousu, J., Jouhten, P., et al. 2008. An analytic and systematic framework for estimating metabolic flux ratios from ^{13}C tracer experiments. *BMC Bioinform.* 9, 266–285.

Raymond, J., and Willett, P. 2002. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *J. Comput. Aided Mol. Design* 16, 521–533.

Riesen, K., Neuhaus, M., and Bunke, H. 2007. Bipartite graph matching for computing the edit distance of graphs. *Graph Based Represent. Patt. Recogn.* 1–12.

Rousu, J., Rantanen, A., Maaheimo, H. et al. 2003. A method for estimating metabolic fluxes from incomplete isotopomer information. *Lect. Notes Comput. Sci.* 2602, 88–103.

Sussenguth, E. 1965. A graph-theoretical algorithm for matching chemical structures. *J. Chem. Doc.* 5, 36–43.

Tarjan, R. 1977. Graph algorithms in chemical computation, 1–20. *In* Christofferson, R., ed. *Algorithms for Chemical Computations*. American Chemical Society, Washington, DC.

Uehara, R., Toda, S., and Nagoya, T. 2005. Graph isomorphism completeness for chordal bipartite graphs and strongly chordal graphs. *Discrete Appl. Math.* 145, 479–482.

Wang, T., and Zhou, J. 1997. EMCSS: a new method for maximal common substructure search. *J. Chem. Inform. Comput. Sci.* 37, 828–834.

Wipke, W., and Dyott, T. 1974. Stereochemically unique naming algorithm. *J. Am. Chem. Soc.* 96, 4834–4842.

Xu, J., 1996. GMA: A generic match algorithm for structural homomorphism, isomorhism, and maximal common substructure match and its applications. *J. Chem. Inform. Comput. Sci.* 36, 25–34.

Yamanishi, Y., Hattori, M., Kotera, M., et al. 2009. E-zyme: predicting potential ec numbers from the chemical transformation pattern of substrate-product pairs. *Bioinformatics* 25, 179–186.

Address correspondence to:
Dr. Markus Heinonen
Department of Computer Science
P.O. Box 68
University of Helsinki
Helsinki 00014, Finland

E-mail: markus.heinonen@cs.helsinki.fi