JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

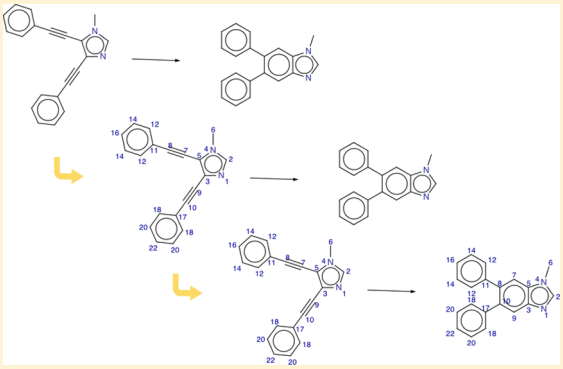
ReactionMap: An Efficient Atom-Mapping Algorithm for Chemical Reactions

David Fooshee,[†] Alessio Andronico,[‡] and Pierre Baldi*[,†]

[†]Institute for Genomics and Bioinformatics and School of Information and Computer Sciences, University of California, Irvine, California 92697, United States

[‡]Université Paris VII, Diderot, 75013 Paris, France

ABSTRACT: Large databases of chemical reactions provide new data-mining opportunities and challenges. Key challenges result from the imperfect quality of the data and the fact that many of these reactions are not properly balanced or atom-mapped. Here, we describe ReactionMap, an efficient atom-mapping algorithm. Our approach uses a combination of maximum common chemical subgraph search and minimization of an assignment cost function derived empirically from training data. We use a set of over 259,000 balanced atom-mapped reactions from the SPRESI commercial database to train the system, and we validate it on random sets of 1000 and 17,996 reactions sampled from this pool. These large test sets represent a broad range of chemical reaction types, and ReactionMap correctly maps about 99% of the atoms and about 96% of the reactions, with a mean time per mapping of 2 s. Most correctly mapped reactions are mapped with high confidence. Mapping accuracy compares favorably with ChemAxon's AutoMapper, versions 5 and 6.1, and the DREAM Web tool. These approaches correctly map 60.7%, 86.5%, and 90.3% of the reactions, respectively, on the same data set. A ReactionMap server is available on the ChemDB Web portal at http://cdb.ics.uci.edu.



INTRODUCTION

Large databases of chemical reactions, such as Beilstein (Reaxys/Elsevier) and SPRESI (InfoChem), create several new data-mining opportunities and challenges. One major opportunity is that by mining these databases, properly configured machine learning algorithms should be able to learn a theory of chemistry and chemical reactivity and be capable of generalizing it to predict the outcome of arbitrary reactions with a host of possible applications.

Before such a formidable task can be attempted, several other problems must first be solved. One fundamental problem, not addressed here, is that these databases are inherently commercial and not readily available to the academic research community for mining purposes. A second problem has to do with the inconsistent quality of the data entered into these databases over the years. Most of the reactions are not balanced and not atom-mapped. This alone creates significant problems for automated machine understanding of chemical reactions and reactivity.

The purpose of this work is to take a significant step toward solving the atom-mapping problem. Plainly stated, in a balanced chemical reaction where all the atoms are accounted for, the atom-mapping problem is the problem of constructing a one-to-one map between all atoms on the left-hand (reactants) side and the right-hand (products) side. When a reaction is unbalanced, it is still possible to consider a partial atom mapping by trying to identify a maximal subset of atoms that can be mapped in one-to-one fashion between the left- and right-hand sides. Most of the time there is a unique solution. In cases involving symmetries, there may be several symmetrically equivalent mappings.

Balancing reactions is likely to require a fairly deep understanding of chemical reactivity. It is possible that a complete solution for the atom-mapping problem may also require a fairly deep understanding of chemical reactivity. Our intuition, however, is that this is not the case for the majority of situations. Accordingly, the goal here is to develop an algorithm that can address the atom-mapping problem on the broadest possible scale without requiring any deep knowledge of chemistry, resorting primarily to topological properties and correlations between the molecular graphs of the reactants and products.

The atom-mapping problem can be formulated in several ways. Because molecules are typically represented by graphs, the atom-mapping problem is, at its core, a question of graph matching. A reaction's reactants, R, and products, P, represent two undirected potentially disconnected graphs. Nodes in R are labeled, while nodes in P are not. We seek a mapping from R to P such that each node in R is assigned to the correct node in P, that is, the assignment must reflect the underlying chemical rearrangement that occurs in the reaction. Although less natural, the atom-mapping problem can obviously be defined also in the reverse direction from products to reactants.

Received: June 2, 2013
Published: October 25, 2013

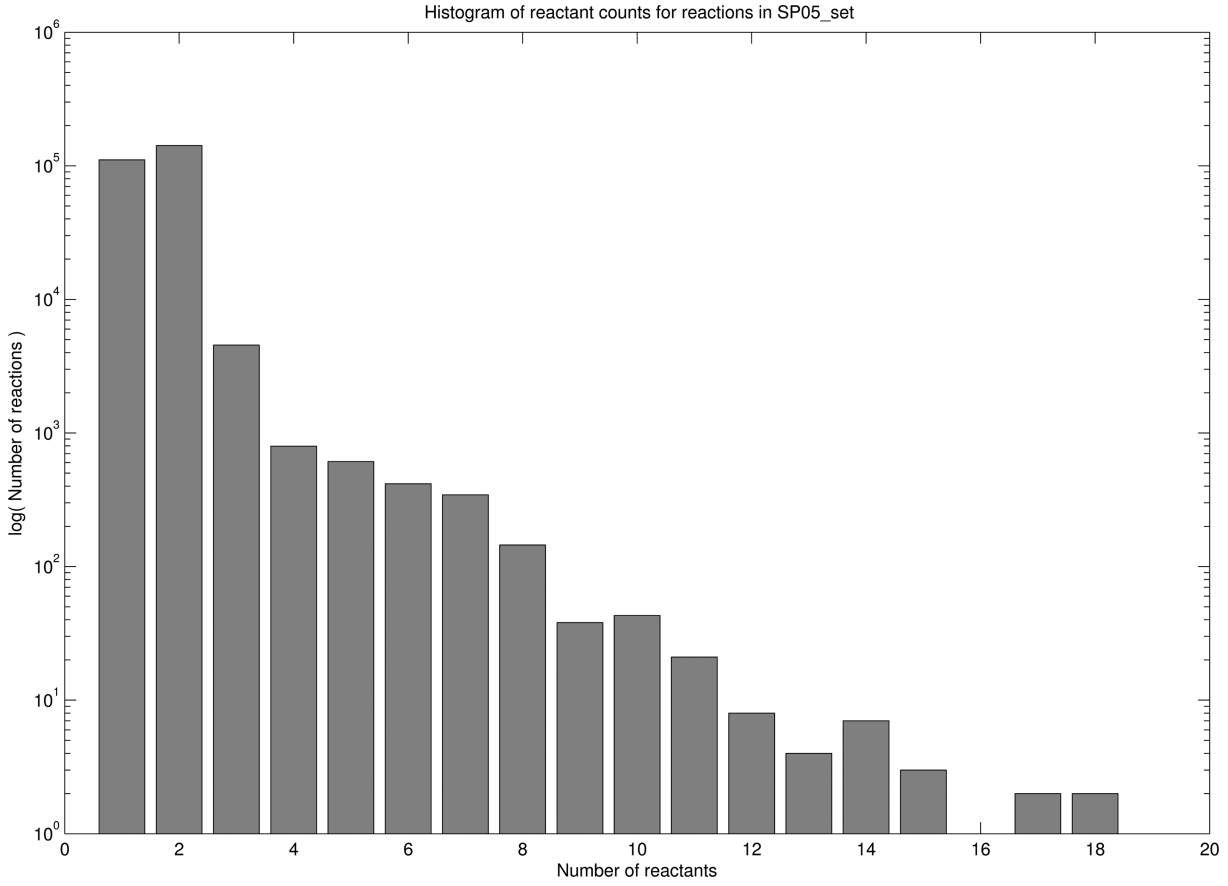


Figure 1. Histogram using a logarithmic scale on the *y*-axis of the number of reactants per reaction computed on the 259,595 reactions from the SP05_set. Most reactions have at most two reactants.

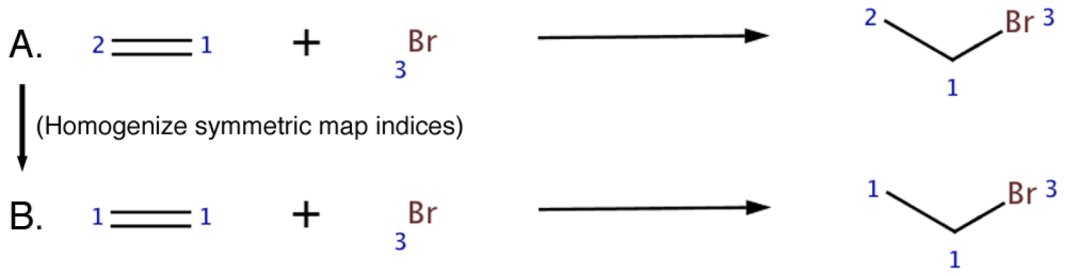


Figure 2. Example of symmetric ambiguity in atom mapping. In A, carbons 1 and 2 are symmetrically equivalent. In B, a homogenization step gives them the same map index, indicating they are interchangeable in the final mapping.

Graph matching is a common task in computer vision and automated object recognition. Prior work addressing the problem includes that of Mjolsness, who described a Lagrangian relaxation network approach.[1] They define a distance measure between adjacency matrices for two graphs and then search for a permutation matrix under constraints defined in the framework of deterministic annealing such that the distance between graphs is minimized. Another approach, described by Taskar et al.,[2] is a convex optimization problem formulation wherein the loss function is a Hamming distance between target mapping and candidate mapping, which simply counts the number of differing variables between the target and candidate solution. Yet another approach involves the use of spectral methods for the permutation group.[3]

Prior work on the atom-mapping problem generally falls under two categories: common substructure-based methods and optimization-based methods.[4] Common substructure methods generate atom mappings by correlating matching subgraphs between reactants and products, followed by additional computation to correlate any remaining atoms that were not part of the common substructure.[5,6] Optimization-based methods attempt to minimize the number of bonds broken and formed during a reaction or to minimize some other cost function over the possible atom mappings for a given reaction.[7−11]

Here, we combine elements of both approaches. First, we identify maximum common subgraphs between the left and right sides of a reaction. Then we search for an assignment of the remaining atoms that minimizes a cost function derived empirically from chemical reaction data.

DATA

To obtain training and testing data, we first removed from the 2005 version of the SPRESI database all reactions that were not

dx.doi.org/10.1021/ci400326p | J. Chem. Inf. Model. 2013, 53, 2812−2819

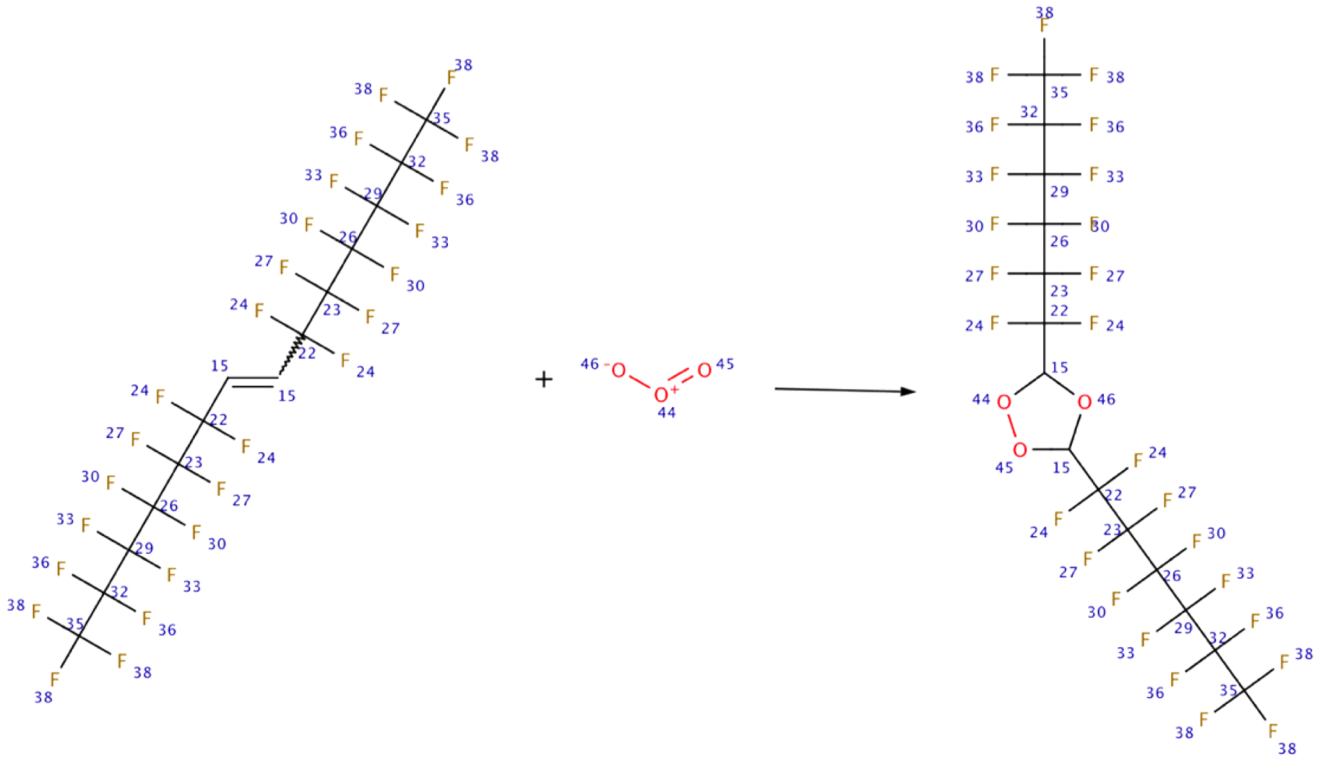


Figure 3. Example of a highly symmetric reaction and its homogenized map indices.

balanced and/or not fully atom-mapped. This resulted in a set of 259,595 reactions (SP05_set). From this set, we extracted two subsets: SP05_test and SP05_training. SP05_test contained 1000 reactions randomly extracted from SP05_set, whereas SP05_training contained the remaining 258,595 reactions. Finally, we also extracted a set, SP09_test, containing 17,996 new balanced and atom-mapped reactions. These reactions were present in the 2009 version of SPRESI but not in the 2005 version. Figure 1 shows the distribution of the number of reactants per reaction in SP05_set.

We used SP05_training to extract first and second neighbors reaction rules as described below. This resulted in 87,371 first neighbors, and 668,684 second neighbors reaction rules. Finally, we tested our approach on both SP05_test and SP09_test.

Database of Reaction Rules. The database of reaction rules is built using the SP05_training data as follows. For each reaction, we extract the lists of atom "state keys" $k^{(1)}(u_i)$ and $k^{(2)}(u_i)$ for reactants, $k^{(1)}(v_j)$ and $k^{(2)}(v_j)$ for products. These are SMILES strings representing the atom u_i with its first neighbor ($k^{(1)}(u_i)$), and first and second neighbors ($k^{(2)}(u_i)$). We then compare the two lists $k^{(1)}(\underline{u})$ and $k^{(1)}(\underline{v})$, and define the first neighbors reaction rules by looking at the differences between the two. Second neighbors reaction rules are similarly defined, using $k^{(2)}(\underline{u})$ and $k^{(2)}(\underline{v})$. For example, the simple addition reaction [CH2:2]=[CH2:1].[BrH:3] ≫ [CH3:2]-[CH2:1][Br:3], shown in Figure 2A, results in the following first neighbors reaction rules: (1) C═C ≫ C[CH2]; rule for the carbon atom with map index 2. (2) C═C ≫ CCBr; rule for the carbon atom with map index 1. (3) Br ≫ [CH2]Br; rule for the bromine atom.

■ ATOM-MAPPING ALGORITHM

Although the algorithm we describe here is not dependent on the particular way a reaction is represented in a computer program, we will use SMILES strings.[12−14] SMILES is a widely used language for representing molecules and reactions as simple text strings. It also provides the foundation for SMARTS and SMIRKS,[15] which allow the efficient representation of molecular patterns (SMARTS) and reaction mechanisms (SMIRKS).

Following the SMILES convention, we will assume that hydrogen atoms not involved in the reaction are implicitly represented, i.e., they are not found in the SMILES string representing the reaction. Moreover, for now, we will only consider balanced reactions, that is, the reactants side will contain the same number and same type of atoms as the products side.

With this in mind, the method we use to construct a one-to-one mapping between reactant and product atoms can be broken down into three steps: (1) Perform a maximum common substructure (MCS) search, and map the atoms belonging to the common substructures that are found. (2) For all atoms that were not mapped in the previous step, apply a bipartite matching with cost function. (3) Return the predicted mapping and any equivalent mappings if requested.

Maximum Common Substructure (MCS) Search (Phase 1). The reactants and the products of a reaction can be considered as graphs where the atoms and bonds represent vertices and edges, respectively. An MCS algorithm can be used to find all the maximal common subgraphs between reactants and products. Corresponding atoms in the common subgraphs are then assigned equal map indices, the assumption being that these portions of the molecules remain unchanged during the reaction.

The maximum common subgraph problem is NP-hard in general, and thus, the application of exact MCS algorithms can be problematic when the molecular graphs contain many nodes. Moreover, not all the atoms in a reaction can be mapped in this way, as the product molecules are the result of structural changes that occur during the reaction (bonds formed and

dx.doi.org/10.1021/ci400326p | J. Chem. Inf. Model. 2013, 53, 2812−2819

broken). Therefore, an MCS algorithm should be supported by other strategies in order to map any remaining atoms. An example of this approach can be found in the article by Leber et al.[16]

In our case, the "exact" MCS algorithm provided by the OEChem library[17] (OpenEye Scientific Software) is attempted for a limited time interval (5 s). This algorithm attempts to find the global maximum common substructure between left- and right-hand sides. If the MCS algorithm returns a match, we continue with phase 2 of the ReactionMap algorithm. Alternatively, if the time interval has elapsed before the exact MCS algorithm completes, it is interrupted, and a second attempt at MCS detection is made using the "approximate" MCS search function. During either of these steps, we may iterate several times in order to recursively find smaller and smaller common substructures in order to match as much of the left- and right-hand sides as possible. Additionally, it is often helpful to relax the specificity of the graph matching employed by the MCS algorithm. That is, we often want to ignore differences in bond order or formal charge in favor of having better total graph coverage from the MCS algorithm. Thus, we configure the MCS algorithm to discern atoms solely based on atomic number (ignoring ring membership, aromaticity, formal charge, and so on), and we configure it to ignore bond order. Once we have a result from the MCS step, we continue with the second phase of the algorithm, in which the remaining atoms must be mapped.

Bipartite Matching (Phase 2). Ideally, the MCS search will return the largest common substructure between reactants and products. The atoms and bonds for which no match is found generally represent "what happened" during the reaction, i.e., the bond rearrangements and consequent atom displacements specific to that particular reaction. On average, after the MCS step, we are left with about 5% of atoms that remain to be atom-mapped.

The approach we use for assigning these atoms is a bipartite matching with cost function. If u_i (v_j) is an atom in the reactants (products) side, we assign a cost $c_{i,j}$ to each of the pairs (u_i,v_j). The problem is then to find the matching that minimizes the total cost of the mapping. In other words, each u_i is assigned to a v_i such that the global cost of the mapping M, $c(M) = \sum_{(i,j)} c_{i,j}$, is the minimum possible. This is the well-known "assignment problem" or "stable marriage problem" and can be efficiently solved with combinatorial optimization techniques. Specifically, we employ the Munkres algorithm,[18] sometimes called the Hungarian method. For this algorithm, the number of operations scales as $O(n^3)$ (n being the number of vertices of the bipartite graph), which results in a polynomial running time. To further increase the speed of the matching, instead of assigning an infinite cost to mapping different atom types (e.g., mapping a carbon atom to a nitrogen), we apply the Munkres algorithm for each atom type separately.

Assignment of Mapping Costs. Mapping costs $c_{i,j}$ are assigned as follows based on the database of reaction rules described above.

For each reactant atom u_i, we create two state keys, $k^{(1)}(u_i)$ and $k^{(2)}(u_i)$. This procedure is repeated for the product atoms v_j to produce $k^{(1)}(v_j)$ and $k^{(2)}(v_j)$. For each pair of atoms (u_i,v_j), we then create the reaction keys (SMILES strings) $r^{(m)}(u_i,v_j) = k^{(m)}(u_i) \gg k^{(m)}(v_j)$ for m = 1,2. Finally, we assign the following mapping costs: (1) If $r^{(2)}(u_i,v_j)$ is in our database of reaction rules, then $c_{i,j}$ = 1. (2) If $r^{(2)}(u_i,v_j)$ is not in our database of reaction rules, but $r^{(1)}(u_i,v_j)$ is in it, then $c_{i,j}$ = 5. (3) If neither $r^{(2)}(u_i,v_j)$ nor $r^{(1)}(u_i,v_j)$ is found in the database, but $k^{(1)}(u_i) = k^{(1)}(v_j)$, then $c_{i,j}$ = 10. (4) If none of the above conditions applies, then $c_{i,j}$ = 100.

In other words, if during training we observed the atom u_i mapped to atom v_j with a corresponding reaction key $r^{(2)}(u_i,v_j)$, then we assign the lowest possible cost (=1). We assign a slightly higher cost (=5) if $r^{(1)}(u_i,v_j)$ was observed, but $r^{(2)}(u_i,v_j)$ was not. A moderate cost (=10) is assigned if neither $r^{(2)}(u_i,v_j)$ nor $r^{(1)}(u_i,v_j)$ was observed, but u_i and v_j are the same atom and have the same first neighbors. The highest cost (=100) is assigned for all other cases. Many different cost values were tested (data not shown), and these were found to work best. In this way, we account for the chemical knowledge extracted during the training phase, while allowing a certain degree of freedom for assigning mappings that were never observed.

Symmetrically Equivalent Mappings. Instead of enumerating and reporting numerous symmetrically equivalent mappings for a given reaction, our approach is to "homogenize" the map indices for symmetrically equivalent atoms. That is, reactant atoms that are symmetrically equivalent are assigned the same map index to indicate their equivalence (Figure 2). After this homogenization step, the atom mapping proceeds normally. In the final proposed mapping, one can see by locating the homogenized atom map indices each possible location where members of a symmetrically equivalent group of atoms could be mapped. Thus, our simple example from above would be reported as [CH2:1]=[CH2:1].[BrH:3] ≫ [CH3:1]-[CH2:1][Br:3]. Map index 2 is omitted, as it was homogenized by assigning an index of 1 to equivalent atoms. This transformation is illustrated in Figure 2, which shows both the original mapping (Figure 2A) and the homogenized version (Figure 2B). We show a more complex example of symmetric mapping in Figure 3, which illustrates an actual mapping from our test set.

If multiple mappings are requested, we can enumerate the symmetrically equivalent outcomes of an atom mapping. Considering our example again, we would in this case return both [CH2:1]=[CH2:1].[BrH:3] ≫ [CH3:1][CH2:1][Br:3] and [CH2:1]=[CH2:2].[BrH:3] ≫ [CH3:1][CH2:2][Br:3], which are the two equivalent mappings for this reaction.

RESULTS

In the previous sections, we outlined the ReactionMap algorithm and the data used to train and test that algorithm. Here, we describe the results obtained from our methods.

Accuracy. Mapping results are summarized in Tables 1 and 2. The full algorithm is able to correctly map all the atoms for 96.2% of the reactions in the test SP05_test set and 95.7% of the

Table 1. Percentage of Correctly Mapped Reactions When Using Different Algorithm Components

algorithm	SP05_test (1000 reactions) (%)	SP09_test (17,996 reactions) (5)
MCS only	78.2	89.7
bipartite matching only	6.2	2.0
combined	96.2	95.7

Table 2. Percentage of Correctly Mapped Atoms When Using Different Algorithm Components

algorithm	SP05_test (1000 reactions) (%)	SP09_test (17,996 reactions) (%)
MCS only	86.3	93.7
bipartite matching only	49.8	44.9
combined	99.4	98.8

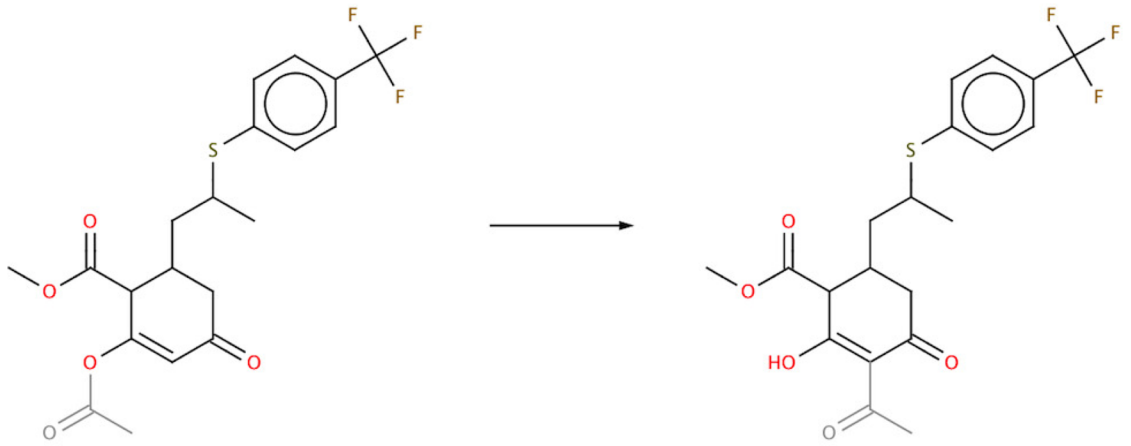


Figure 4. Correctly mapped reaction illustrating both maximum common substructure (MCS) search and bipartite matching steps. The gray region indicates atoms that were not mapped during the MCS step but were correctly mapped during the bipartite matching step.

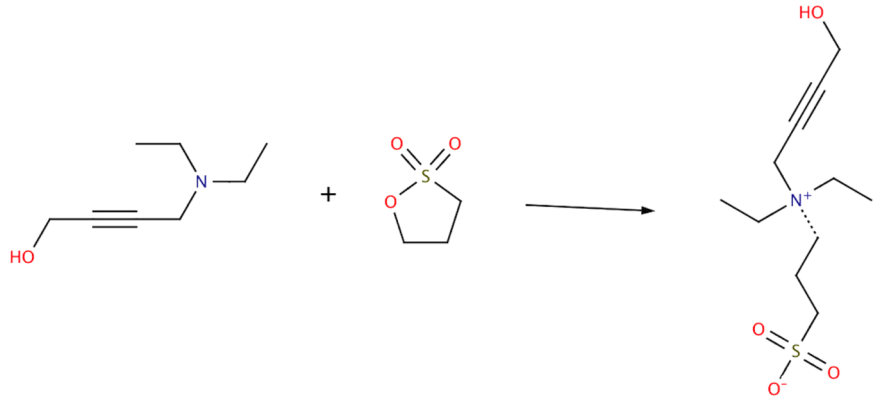


Figure 5. Correctly mapped reaction that requires only the MCS step to map all atoms. The dotted line indicates the new bond formed between the two reactant molecules.

Table 3. Percentage of Reactions Correctly Mapped in Reverse

SP05_test (1000 reactions) (%)	SP09_test (17,996 reactions) (%)
94.4	95.0

reactions in the SP09_test with a mean time per mapping of 2 s. For the remaining reactions, a subset of the atoms is still mapped correctly, on average about 80%. Thus, the percentage of correctly mapped atoms across all the reactions is 99.4% for the SP05_test set and 98.8% for the SP09_test set. These percentages are probably a slight underestimate of what can be achieved by our methods because there are probably some errors in SPRESI affecting both the training and testing procedures.

The tables also show the effect of each component of the algorithm on overall performance and why each step is necessary. MCS alone, for instance, can only map about 78.2% of the reactions in the SP05_test set, and the bipartite matching alone can only map 6.2% of the reactions in the same set. Within each component, we also observe a greater variability across data sets. For instance, MCS alone can map 89.7% of the reactions contained in the SP09_test set. The origin of this variability may be in part explained by fluctuations in the composition of these two data sets. For example, the percentages of reactions with exactly one reactant and exactly two reactants are 42% and 54%, respectively, in the SP05_test set, whereas these percentages are equal to 50% and 45.5%, respectively, in the SP09_test set. In any case, the MCS component is responsible for most of the accuracy with a small but significant additional improvement from the bipartite matching component.

Table 4. Percentage of Atoms in Agreement between Forward and Reverse Mapped Reactions

result	SP05_test (1000 reactions) (%)	SP09_test (17,996 reactions) (%)
correct mapping	99.6	99.7
incorrect mapping	83	76.2

ReactionMap succeeds at mapping a wide variety of reactions, including many involving large complex molecules and structural rearrangements. Figure 4 illustrates a successful mapping that requires both the MCS step and the bipartite matching of the remaining atoms. While many reaction mappings require a combination of MCS and bipartite matching, some can be performed using MCS alone. Figure 5 illustrates one such situation. Here, the two reactants bond together to form a single product. Because there is no structural rearrangement of either reactant, this is an ideal situation for MCS to map the full reaction.

Additionally, we can perform the mapping in the reverse direction, matching products to reactants. For this, we use first and second neighbor rules extracted as described above but in the reverse direction. Table 3 summarizes the percentage of reactions mapped correctly in reverse. We tried many different ways of combining the additional information from the reverse

dx.doi.org/10.1021/ci400326p | J. Chem. Inf. Model. 2013, 53, 2812−2819

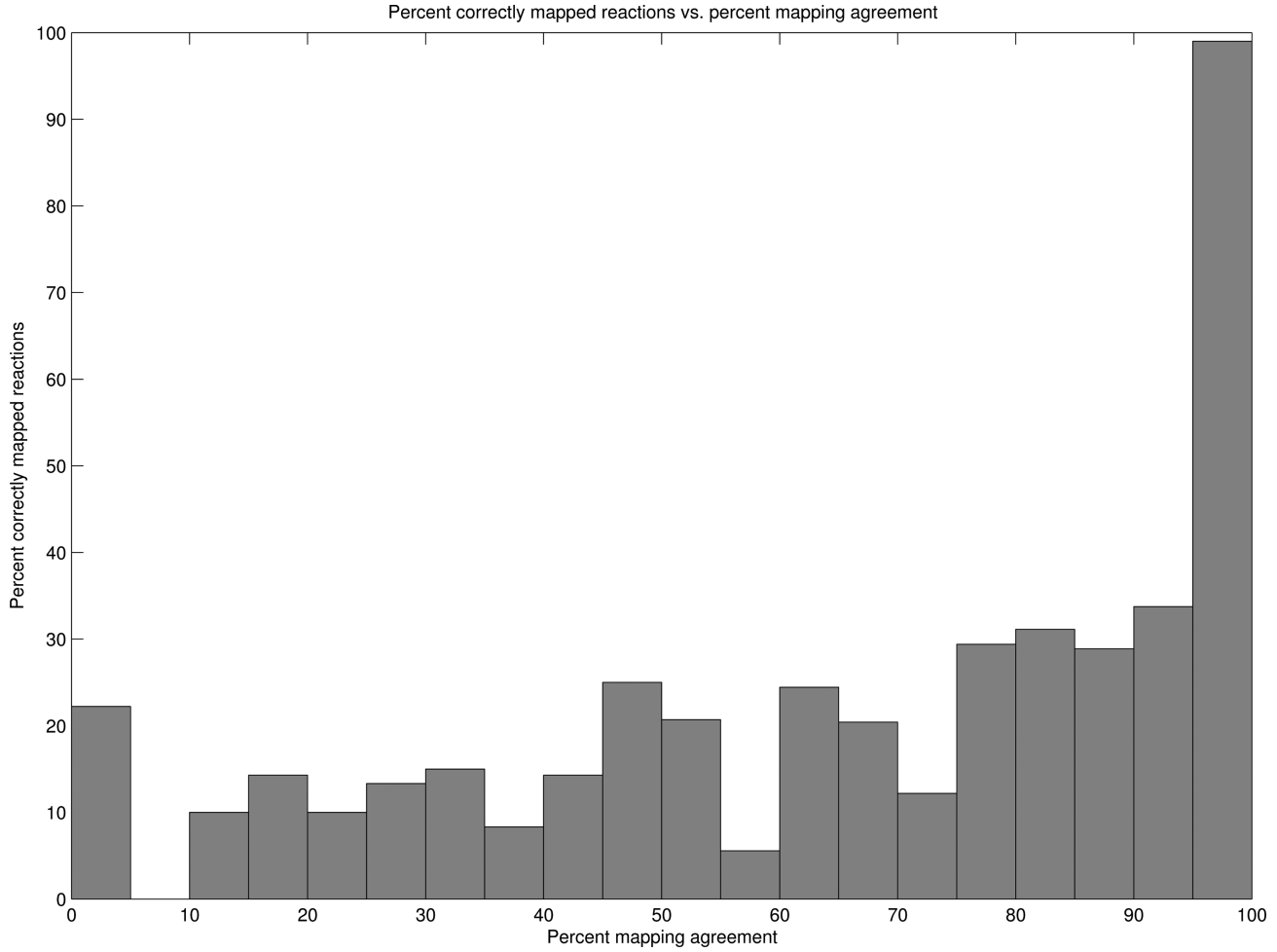


Figure 6. Histogram showing the percentage of correct mappings for various degrees of forward−reverse agreement. For agreement greater than 95%, 99% of mappings are correct, while lower levels of agreement have significantly fewer correct mappings on average.

Table 5. Comparison with Other Predictors

algorithm	time per mapping (s)	SP05_test (1000 reactions) (%)
ReactionMap	2	96.2
AutoMapper 5	0.03	60.7
AutoMapper 6.1	0.02	86.5
DREAM	<2	90.3

mappings to complement forward mappings, but none led to a robust performance improvement. Nonetheless, performing the reverse mappings is still useful, as we can check the agreement between the two mappings and gain an idea of our confidence in the final mapping. Table 4 summarizes the average percentage of atoms in agreement for correct and incorrect mappings. Specifically, incorrect mappings have on average 83% of atoms matching between the suggested forward and reverse mappings. For correct mappings, this agreement averages 99.6%. Figure 6 shows the average mapping success rate versus the percent agreement between forward and reverse mappings for SP09_test. We find that 99% of mappings are correct when agreement is above 95%. For lower levels of agreement, the likelihood of a correct mapping declines rapidly. Additionally, of the 96.2% (SP05_test) and 95.7% (SP09_test) of reactions mapped correctly, 98.1% and 98.9% of these, respectively, have greater than 95% forward−reverse agreement. Thus, the majority of correctly mapped reactions are mapped with high confidence.

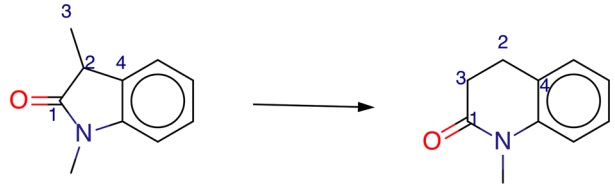


Figure 7. Example of an incorrectly mapped reaction due to an error during the MCS step. The correct mapping involves breaking the bond between carbons 1 and 2, followed by bond formation between carbons 1 and 3. Instead, the MCS step returned a mapping that broke bond 2−4 and joined carbons 3 and 4.

Comparison with Other Predictors. We compared our results on SP05_test with the performance of three publicly available atom-mapping tools on that same test set (Table 5). The software tested included ChemAxon's AutoMapper,[19] versions 5 and 6.1, and the DREAM Web tool described by First et al.[11] AutoMapper uses an MCS approach, while DREAM takes an optimization-based approach, using linear programming to minimize bonds broken and formed. AutoMapper versions 5 and 6.1 mapped 60.7% and 86.5% of reactions correctly, respectively, while DREAM mapped 90.3% of the reactions correctly. ReactionMap correctly mapped 96.2% of the reactions.

ReactionMap's average mapping speed on SP05_test was 2 s per reaction. AutoMapper appears to sacrifice mapping accuracy for

dx.doi.org/10.1021/ci400326p | J. Chem. Inf. Model. 2013, 53, 2812−2819

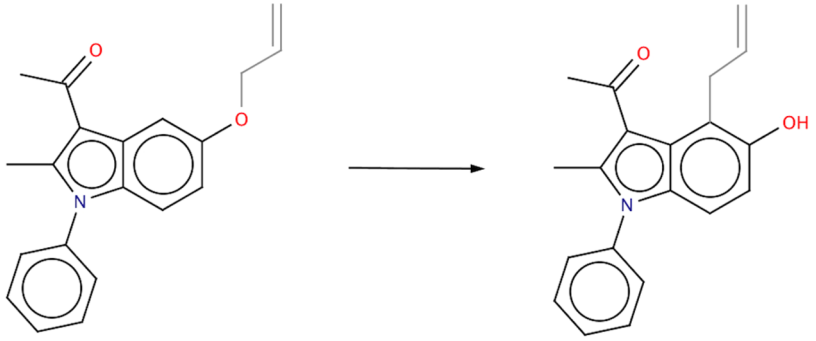


Figure 8. Example of an incorrectly mapped reaction due to an error during the bipartite matching step. The gray region indicates atoms that were not mapped during the MCS step. Bipartite matching of these atoms returned the incorrect (flipped) mapping order.

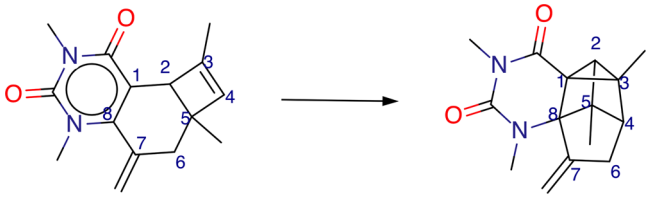


Figure 9. Example of a problematic reaction that was mapped incorrectly. The challenge stems from interdependent or sequential bond breaking and formation. Here, bond 5−6 must first be broken to allow formation of bonds 1−3, 8−5, and 4−6.

speed, with average mapping times of 0.02−0.03 s per reaction. Because we could only interact with DREAM through a Web interface, rather than running tests locally, a direct speed comparison is not available. However, we estimate the mapping time to be slightly less than 2 s per reaction on this data set based on turnaround time from the DREAM Web service.

Error types. Errors made by the algorithm are a function of two factors: how many atoms the MCS search maps between reactants and products and how well the bipartite matching performs on the remaining unmapped atoms. Errors can occur during either step. Figure 7 shows the case where an error occurs during the MCS step. Here, bond 1−2 breaks, and a new bond forms between carbons 1 and 3. The MCS step is able to find a common substructure which includes all atoms and all bonds except bond 2−4. Thush it returns a mapping that breaks bond 2−4, and joins carbons 3 and 4. The second possibility—an error during the bipartite matching step—is shown in Figure 8. In this example, a rearranged propene group is mapped with its indices flipped. Figures 9 and 10 show examples of reactions that can be problematic because of interdependent or sequential bond breaking and formation. In Figure 9, the bond between carbons 5 and 6 must be broken to allow formation of bonds 1−3, 8−5, and 4−6. Figure 10 shows a similar situation—the bond between sulfur 1 and carbon 2 must be broken, with the cyclopentadiene rejoining the two intermediates. When given the three reactants present after bond 1−2 is broken (Figure 11), the algorithm produces the correct mapping.

Though a handful of errors do occur as outlined above, ReactionMap's mapping success rate of about 96% at the reaction level and 99% at the atom level is encouraging considering the huge variety and complexity of reactions tested and possible directions for future improvements, such as using larger training sets. In future work, mapping accuracy could be improved by developing additional heuristics or by harnessing additional empirical chemical data to address error cases like those described above. The algorithm could also be further optimized for unbalanced reactions and thus should become

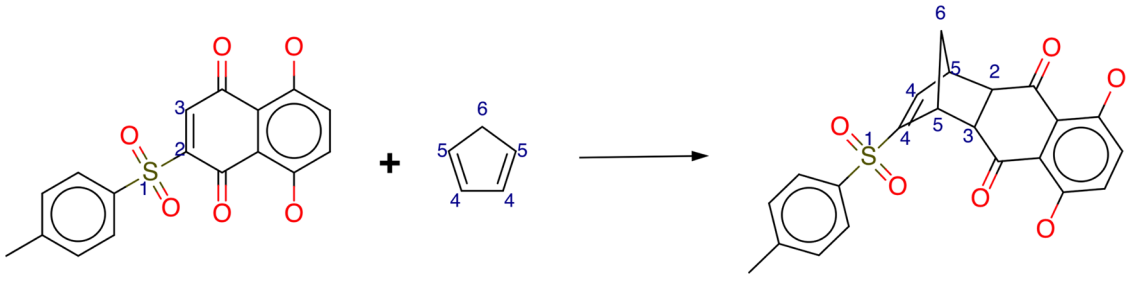


Figure 10. Example of another problematic reaction that was mapped incorrectly. Bond 1−2 is broken, yielding two fragments, which are rejoined by the cyclopentadiene. If the algorithm is asked to map the three reactants after bond 1−2 is broken, the result is correct.

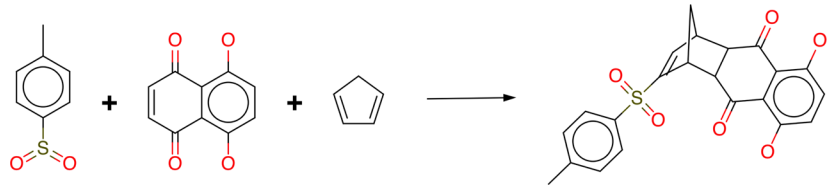


Figure 11. Reaction after breaking bond 1−2 in Figure 10. From this point, the reaction is mapped correctly, illustrating the challenge of interdependent bond breaking and bond formation.

dx.doi.org/10.1021/ci400326p | J. Chem. Inf. Model. 2013, 53, 2812−2819

useful in time as a tool for filtering and cleaning large reaction databases, although these remain largely unavailable for mining purposes. In any case, ReactionMap represents a strong tool for solving the atom-mapping problem and opening the door for using more completely atom-mapped chemical data sets for future machine learning and other chemoinformatics endeavors. A ReactionMap server is available on the ChemDB Web portal at http://cdb.ics.uci.edu.

AUTHOR INFORMATION

Corresponding Author

*E-mail: pfbaldi@ics.uci.edu.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

Work was supported in part by Grants NSF IIS-0513376, NIH LM010235, and NIH NLM T15 LM07443 to P.B. We wish to acknowledge OpenEye Scientific Software and ChemAxon for academic software licenses and Jordan Hayes and Yuzo Kanomata for computing support.

REFERENCES

(1) Rangarajan, A.; Mjolsness, E. D. A Lagrangian relaxation network for graph matching. *IEEE Trans. Neural Networks* **1996**, *7*, 1365−1381.
(2) Taskar, B.; Chatalbashev, V.; Koller, D.; Guestrin, C. Learning Structured Prediction Models: A Large Margin Approach, 2005. http://doi.acm.org/10.1145/1102351.1102464 (accessed June 1, 2013).
(3) Huang, J.; Guestrin, C.; Guibas, L. Fourier theoretic probabilistic inference over permutations. *J. Mach. Learn. Res.* **2009**, *10*, 997−1070.
(4) Chen, W. L.; Chen, D. Z.; Taylor, K. T. Automatic reaction mapping and reaction center detection. *WIREs Comput. Mol. Sci.* **2013**, *3* (6), 560−593.
(5) McGregor, J. J.; Willett, P. Use of a maximum common subgraph algorithm in the automatic identification of ostensible bond changes occurring in chemical reactions. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 137−140.
(6) Apostolakis, J.; Sacher, O.; Körner, R.; Gasteiger, J. Automatic determination of reaction mappings and reaction center cnformation. 2. Validation on a biochemical reaction database. *J. Chem. Inf. Model.* **2008**, *48*, 1190−1198.
(7) Akutsu, T. Efficient Extraction of Mapping Rules of Atoms from Enzymatic Reaction Data, 2003. http://doi.acm.org/10.1145/640075.640076 (accessed November 2, 2013).
(8) Crabtree, J. D.; Mehta, D. P. Automated reaction mapping. *J. Exp. Algorithmics* **2009**, *13*, 15:1.15−15:1.29.
(9) Heinonen, M.; Lappalainen, S.; Mielikäinen, T.; Rousu, J. Computing atom mappings for biochemical reactions without subgraph isomorphism. *J. Comput. Biol.* **2011**, *18*, 43−58.
(10) Latendresse, M.; Malerich, J. P.; Travers, M.; Karp, P. D. Accurate atom-mapping computation for biochemical reactions. *J. Chem. Inf. Model.* **2012**, *52*, 2970−2982.
(11) First, E. L.; Gounaris, C. E.; Floudas, C. A. Stereochemically consistent reaction mapping and identification of multiple reaction mechanisms through integer linear optimization. *J. Chem. Inf. Model.* **2012**, *52*, 84−92.
(12) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.
(13) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97−101.
(14) Weininger, D. SMILES. 3. Depict. Graphical depiction of chemical structures. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 237−243.
(15) Daylight. www.daylight.com (accessed January 1, 2013).
(16) Leber, M.; Egelhofer, V.; Schomburg, I.; Schomburg, D. Automatic assignment of reaction operators to enzymatic reactions. *Bioinformatics* **2009**, *25*, 3135−3142.
(17) *OEChem*, version 1.7.4; OpenEye Scientific Software, Inc. www.eyesopen.com (accessed November 2, 2013).
(18) Munkres, J. Algorithms for the assignment and transportation problems. *J. Soc. Ind. Appl. Math.* **1957**, *5*, 32−38.
(19) *Marvin Beans*, version 6.1; ChemAxon Ltd. www.chemaxon.com (accessed November 2, 2013).