Universal Structure Conversion Method for Organic Molecules: From Atomic Connectivity to Three-Dimensional Geometry

Yeonjoon Kim and Woo Youn Kim*

*Department of Chemistry, KAIST, Daejeon 305-701, Korea. *E-mail: wooyoun@kaist.ac.kr*

Received February 4, 2015, Accepted March 5, 2015, Published online June 14, 2015

We present a powerful method for the conversion of molecular structures from atomic connectivity to bond orders to three-dimensional (3D) geometries. There are a number of bond orders and 3D geometries corresponding to a given atomic connectivity. To uniquely determine an energetically more favorable one among them, we use general chemical rules without invoking any empirical parameter, which makes our method valid for any organic molecule. Specifically, we first assign a proper bond order to each atomic pair in the atomic connectivity so as to maximize their sum and the result is converted to a SMILES notation using graph theory. The corresponding 3D geometry is then obtained using force field or *ab initio* calculations. This method successfully reproduced the bond order matrices and 3D geometries of 10 000 molecules randomly sampled from the PubChem database with high success rates of near 100% except a few exceptional cases. As an application, we demonstrate that it can be used to search for molecular isomers efficiently.

Keywords: Adjacency matrix, Bond order assignment, SMILES, Graph theory

Introduction

Structural information about molecules is a key ingredient in chemical information and modeling. There are various ways to represent molecular structures. If we regard a molecule as the collection of atoms connected via chemical bonds, its structural information can be classified into the following three levels in the order of decreasing complexity: three-dimensional (3D) geometry, bond orders (BOs), and atomic connectivity (AC). Each level can be used for different purposes. For instance, the 3D geometry uniquely determines the Hamiltonian of a molecule and thus enables us, in principle, to know all the electronic information of the molecule with help from *ab initio* quantum chemistry. In molecular mechanics simulations, the BO information is necessary to select appropriate force field parameters.[1] Chemical reactions can be understood as the successive formation and dissociation of chemical bonds between molecules. The AC information will be sufficient to describe such new bond formation and dissociation.[2–5]

Although 3D geometry implies full information about molecular structures, in some cases, it may cause unnecessary complexity in computational handling of molecular structures due to redundant information. Therefore, an optimal level of information should be chosen to simplify problems as well as to enhance computational efficiency, which naturally demands the development of a structure-conversion method from one level of information to another. Conversion from a higher level to a lower level is trivial, whereas there is in general no one-to-one mapping between levels for opposite cases. Thus, additional information is necessary to uniquely determine a molecular structure at a higher level converted from a lower level. However, it is apparently possible to make one-to-one mapping between most stable structures at different levels. We here aim to develop a structure-conversion method via one-to-one mapping between two structures without system-dependent additional information. To obtain the most stable structures at each level, we utilize universal chemical information such as valence rules.

First, we assign BOs to each atomic pair of a given AC. Typically, BOs are determined by hybridization analysis using bond lengths and angles[6,7] or by comparison of atomic pairs with a known database.[8,9] They can be also assigned using chemical or length rules.[10] However, those methods demand knowledge of the 3D geometry of the molecules. Although the BOs can be assigned using valence rules without 3D coordinates,[1,11] such rules have many exceptions and thus often demand careful post-processing. Wang *et al.* proposed a practical and reliable method to find BOs directly from AC information.[1] They use a trial and error approach for all the possible valence states of atoms in a molecule and select an optimal state by comparing the penalty scores of each candidate; the penalty scores are measured from preassigned values as an input. However, an empirical parameter-free method for wide applications has yet to be made available.

Once the BO assignment is completed, a SMILES string[12–14] can be generated using the BO information prior to building a final 3D structure. The SMILES string has been widely used because it is simple, flexible, and still informative: *e.g.*, it even distinguishes *R*/*S* and *E*/*Z* isomers in a plain notation. Thus, it has been regarded as the best symbol method to encode a molecular connection table.[15,16] Graph search algorithms can be used to write a SMILES string from the BO information. For example, rings in a molecule can readily be detected by a mathematical algorithms used for finding the smallest set of smallest rings (SSSR) from the chemical graph representation of the molecule.[17–19] We found that the

Kruskal algorithm is particularly useful for the ring detection; this algorithm was originally used to find a minimum spanning tree (MST).[20] A main chain and branches of molecules can be found using the depth first search (DFS) algorithm.[21] The breadth first search (BFS) algorithm[22] can be used to find stereogenic centers. Then, the SMILES string is readily converted to a 3D geometry using a program such as OpenBabel,[23,24] and the resulting structure can be further cleaned up using an additional force field or ab initio calculations if necessary.

Herein, we present an efficient and universal method for obtaining 3D structures of organic molecules solely from AC information. In what follows, we explain the numerical methods used at each transformation step from the AC of a molecule to the BOs to the final 3D structure. Then, we assess the accuracy of our method by applying it to 10 000 molecular structures randomly sampled from the PubChem database.[25–27] Finally, we demonstrate its usefulness via a very efficient search for molecular isomers.

Methods

Conversion from AC to BOs. For a given AC, one can assign several plausible sets of BOs (*e.g.*, Figure 1(a)). Among these, we can select an energetically more favorable one or ones according to, *e.g.*, chemical rules, as BOs are directly related to molecular stability. In this process, we use only information on the valence and possible formal charges of atoms with universal selection criteria: maximum BOs and charge conservation (Figure 2).

We first assign the valence of atoms (N_v) as an input. Most atoms have unique valences, but some of them such as N, O, P, and S can also have two or more values depending on the number of adjacent atoms. Table 1 shows the values of N_v of the atoms considered in this work. Then, we need to know the degree of unsaturation (DU) of each atom in order to assign multiple bonds between unsaturated atoms (UAs). The DU of each atom is readily calculated from the AC of a molecule or its matrix version, namely, an adjacency matrix (**A**) or its element (A_{ij}), as follows:

$$u(i) = N_{v,i} - \sum_j A_{ij} \tag{1}$$

where i and j denote atom indices and $u(i)$ and $N_{v,i}$ are the DU and valence, respectively, of atom i. Atoms with a positive DU are saved in a UA list. If m atoms have two N_v values, all the possible valence states (2^m sets) are considered and then these states are sorted in decreasing order of the sum of DUs to generate a set of BOs having more multiple bonds earlier. If all the adjacent atoms of a UA are saturated, the UA cannot form multiple bonds any more, and so it is removed from the UA list.

Before assigning multiple bonds to the UAs, BOs of the following exceptional case are predetermined: carbon monoxide has a triple bond. Subsequently, the BO assignment begins for the remaining atoms. We note that the resulting set of BOs strongly depends on the starting point and direction of the assignment, but searching all the sets of BOs is practically inefficient. Therefore, the choices of an appropriate starting point and direction are crucial to finding an optimal set of BOs in the early stage of searching. Figure 1 shows an example of a biphenyl molecule. In Figure 1(a), as the left molecule has six double bonds, while the right one has five, the former is energetically preferred to the latter. Figure 1(b) illustrates the process of obtaining the left molecule. In a UA list, an atom with the least number of adjacent atoms is chosen as a starting point, because it has fewer choices with which to make an atomic pair. The filled circle in Figure 1(b) is randomly chosen among the 10 equally probable atoms that all have two adjacent atoms. Likewise, among the adjacent atoms of the starting atom, we choose one that has the lowest number of adjacent atoms. Then, we assign a multiple bond between the two atoms and, accordingly, their DUs and the UA list are updated. This procedure is repeated until no atom with a positive DU is left, which completes one BO matrix (**B**). We determine whether the resulting BO matrix satisfies the following conditions:

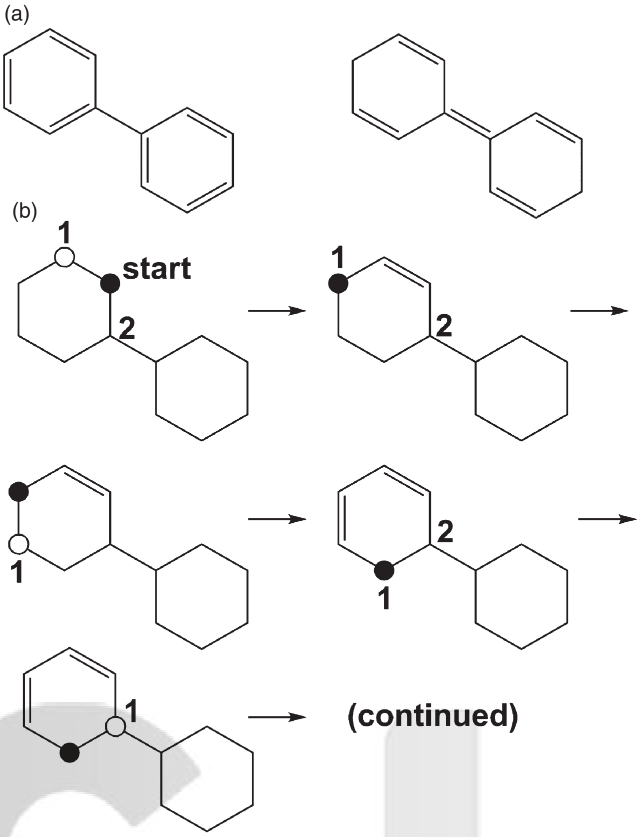
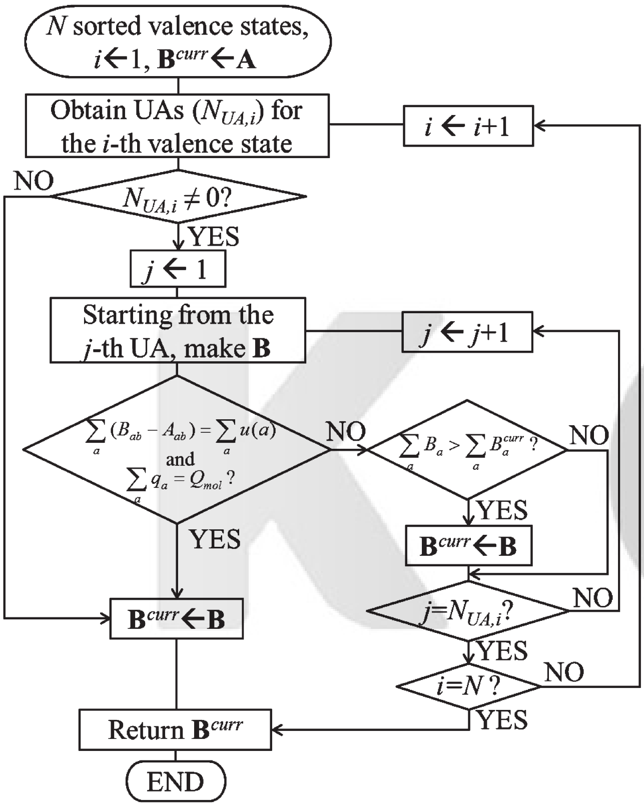


Figure 1. (a) Examples of BO assignment for a biphenyl molecule. (b) Step-by-step process of BO assignment. The filled and hollow circles indicate the current and next atoms to be paired, respectively. The integer values indicate the number of adjacent atoms for a given atom.

$$\sum_{ab} (B_{ab} - A_{ab}) = \sum_a u(a) \text{ and } \sum_a q_a = Q_{\text{mol}} \tag{2}$$

where a and b represent the atoms in a molecule, q_a denotes the formal charge of atom a, and Q_{mol} is the total molecular charge. The first and second conditions are performed to check whether the UAs are still left and whether the total charge is conserved, respectively. The formal charges of each atom are calculated using the rules shown in Table 2. If the two conditions are not satisfied, we need a new BO assignment starting from a different atom. Eventually a BO matrix that either satisfies the above conditions or has the maximum sum among all the resulting BO matrices is selected. Figure 2 shows the flow chart for this procedure.



N sorted valence states, $i \leftarrow 1$, $\mathbf{B}^{curr} \leftarrow \mathbf{A}$
Obtain UAs ($N_{UA,i}$) for the i-th valence state
$i \leftarrow i+1$
NO
$N_{UA,i} \neq 0$?
YES
$j \leftarrow 1$
Starting from the j-th UA, make $\mathbf{B}$
$j \leftarrow j+1$
$\sum_a (B_{ab} - A_{ab}) = \sum_a u(a)$ and $\sum_a q_a = Q_{mol}$?
NO
$\sum_a B_a > \sum_a B_a^{curr}$?
NO
YES
$\mathbf{B}^{curr} \leftarrow \mathbf{B}$
YES
$\mathbf{B}^{curr} \leftarrow \mathbf{B}$
$j=N_{UA,i}$?
NO
YES
$i=N$?
NO
YES
Return $\mathbf{B}^{curr}$
END

Figure 2. Flow chart of BO assignment.

Table 1. Atomic valences.

Elements	N_v	Elements	N_v
H	1	F, Cl, Br	1
B	3	N	3 or 4
C	4	P	3, 4, or 5
O	1 or 2	S	2, 4, or 6

If all the atoms in a given AC have single valences, the BO assignment is terminated with the BO matrix obtained through the above procedure. Otherwise, we restart the algorithm with a different set of atomic valences and repeat it until the above termination conditions (Eq. (2)) are satisfied. If all the valence states fail to satisfy the conditions, we choose the BO matrix whose sum is the largest, as described in Figure 2. We emphasize that as the BO assignment begins with a valence state that has the largest sum of DUs, once the conditions in Eq. (2) are satisfied, the BO matrix obtained at an earlier step automatically has the largest sum.

Conversion from BOs to SMILES. To generate a SMILES string from the selected BO matrix, we follow the Kekule convention, which does not need to distinguish aromatic bonds from other carbon–carbon double bonds (C═C). Kekule notation needs to be specified a main chain and branches of a molecule and also so-called ring closure atoms if the molecule includes a ring or rings. In the following, we first search for ring closure atoms of a molecule using the Kruskal algorithm from graph theory; then, we detect the main chain and its branches using the DFS algorithm.

We explain how to find ring closure atoms using a coronene molecule as an example. For the sake of convenience, all the terminal hydrogens are omitted. Then, we represent the molecular structure with a graph and detect its MST using the Kruskal algorithm.[20] The bold line in Figure 3 shows an example of the MST for coronene. As the MST by definition connects all the atoms without making rings, the remaining bonds, which are not included in the MST (dashed lines in Figure 3), naturally specify ring closure atoms. To perform the Kruskal algorithm, we need to define the weights of each edge. We note that in Kekule notation, the BO between ring closure atom pairs should be single. Therefore, the weight of each edge is given as the inverse of its corresponding BO (*e.g.*, 1.0 for singles, 0.5 for doubles, and 0.33 for triples) in order to cause the MST to include multiple bonds with higher priority.

To find a main chain and branches from the MST, we use the nonrecursive DFS algorithm.[21] Figure 4 shows how the main chain and the root atoms of each branch are detected by analyzing the change in stack elements. The main chain starts from the first atom in the molecular graph, labeled as 1, and the atom is pushed into the stack. As the status of DFS propagates along the graph, atoms in the stack and the parent stack are updated accordingly: the "parent" stack contains parent atoms of the atoms in the current "stack." At each status update, an old atom is popped out from the stack and/or a new atom is pushed into it. That the latter case occurs means that a chain propagation is being continued, so that all the

Table 2. Formal charges of atoms in various valence states.

Valence states	Formal charge	Valence states	Formal charge
Hexavalent sulfur	0	Carbon with three single bonds	1/−1 depending on the total charge
Pentavalent phosphorus	0	Boron	3 – (no. of bonds)
Carbon with two single bonds	0	The rest	(no. of valence electrons) – 8 + (no. of bonds)

atoms passed by in this way consist of the chain. In contrast, if no new atom is pushed into the stack, a new branch begins propagating. In Figure 4, branching is found to occur twice at statuses M6 and B9. The root atoms of the new branches are determined by the entries of the corresponding parent stack as denoted by the underlined numbers in Figure 4. This procedure is repeated until all the atoms in the graph are scanned.

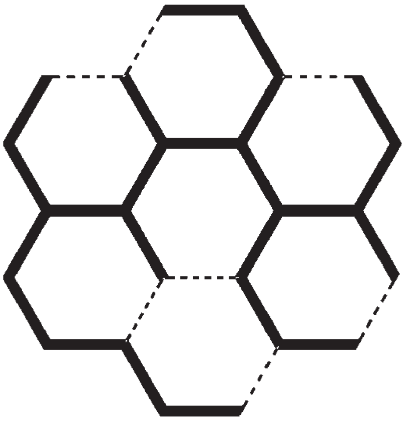
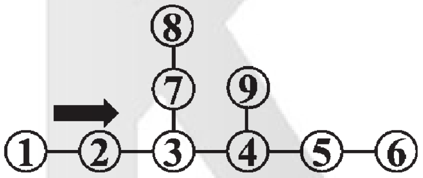


Figure 3. Minimum spanning tree (bold line) of a coronene molecule obtained from the interatomic weights assigned by the inverse values of bond orders. The dotted lines indicate chemical bonds between ring closure atoms.



Status	Stack	Parent
Start	1	Null
M1	2	1
M2	3	2
M3	4,7	3,3
M4	5,9,7	4,4,3
M5	6,9,7	5,4,3
M6	9,7	4,3
B9	7	3
B7	8	7
B8	Null	Null

Figure 4. Process of searching for the main chain and branches of a molecular graph using the DFS algorithm. The entries of the stack and of the parent columns represent atoms and their parent atoms in the graph at each status of searching, respectively. M and B indicate the main chain and the branch, respectively. The underlined elements in the parent stack correspond to the root atoms of the branches.

One of the useful features of SMILES is that it is able to consider stereochemistry. To incorporate this feature into our method, we need to detect the chiral or *E*/*Z* centers of a given molecule from its BO information. Basically, chiral or *E*/*Z* centers have four branches. Therefore, using a BO matrix, we first detect tetravalent atoms or C═C bonds that do not belong to a ring; we then consider them as candidate chiral or *E*/*Z* centers, respectively. To determine whether their four branches are different from each other, we decompose the molecule into four separate pieces, including each branch, and then compare them with each other. This decomposition can be done by removing the corresponding center-atom or C═C bonds in the adjacency matrix of the target molecule.

If some of branches are connected with others, we separate them using the BFS algorithm. Figure 5(a) shows an example of such a process for the detection of a chiral center. After removing the center atom, the molecule is decomposed into three groups: H, Br, and the remainder. Two branches belonging to the third group are connected with one another. The BFS simultaneously begins propagating from both ends of the third group, as denoted by the arrows in Figure 5(a). Once the propagation reaches the same atom, that atom is removed, giving rise to two separate groups. In the case of *E*/*Z* centers, we simply replace the chiral center by the C═C bond as an *E*/*Z* center (Figure 5(b)). Finally, we need to make a comparison among

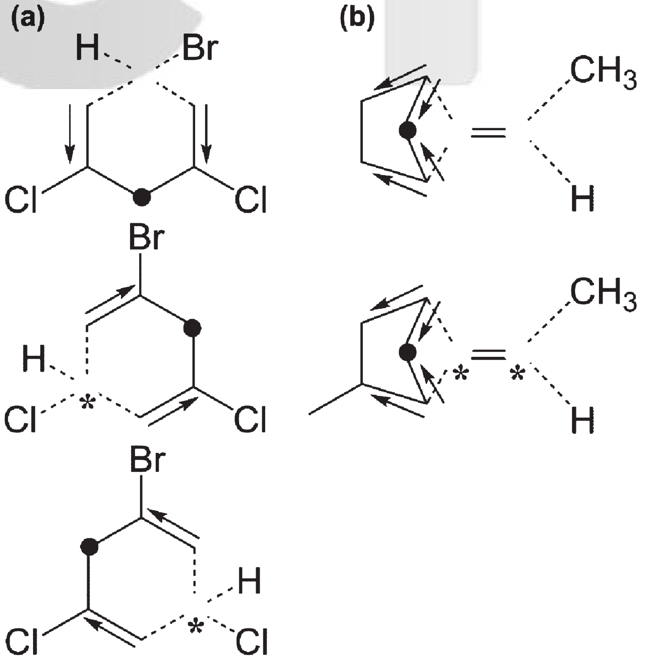


Figure 5. Schematic drawing of the BFS algorithm for detecting (a) chiral and (b) *E*/*Z* centers. The asterisks mean that a chiral or *E*/*Z* center atom at the given position was deleted; the dotted lines denote chemical bonds between the center atom and its adjacent atoms. The arrows indicate the direction of BFS. The filled circles indicate an atom where a search along two directions meets.

the four decomposed groups. To this end, we compare the alternative Coulomb matrices $\mathbf{C}^A$ of each group[5]:

$$C_{ij}^A = \begin{cases} A_{ij}\cdot Z_i\cdot Z_j & \text{if } i \neq j \\ Z_i^2 & \text{otherwise} \end{cases} \qquad (3)$$

where Z_i is the atomic number of atom i. However, this process is not sufficient to convincingly judge whether the groups are identical, because different numberings of atoms for the same molecule result in different adjacency matrices, as depicted in Figure 6. To distinguish such permutational isomers, we rather compare the eigenvalues of the coulomb matrices, as the permutational isomers share the same eigenvalues.[5] We note that a previous study also reported a detection method for chiral centers in a molecular graph but with slightly different algorithms.[28] That method directly compares each atom of the four branches while propagating atom-by-atom from a chiral center, which essentially gives results identical to ours.

Collected information on formal charges, ring closure atoms, main chain, branches, terminal hydrogens, and stereochemistry, as determined above, is finally used to write a SMILES string according to the known rules.[12]

Conversion from SMILES to 3D Geometry. The 3D geometry of a molecule is generated from the resulting SMILES string. Conventional molecular builders use numerical, rule-based, or data-based methods.[29] We use the OpenBabel program,[23] which adopts both numerical and rule-based methods. It first generates 3D atomic coordinates from SMILES and then optimizes those coordinates using a force field method. We found that for complex molecules it often produces distorted structures, even with steric clashes between bulky groups. Therefore, we optimize the geometry again using the universal force field (UFF) method,[30] as implemented in the Gaussian 09 program suite,[31] in which we apply the QEq charges[32] with fixed connectivity and BOs. Figure 7(a) and (b) shows the geometries of fullerene obtained from OpenBabel and the UFF reoptimization, respectively. If necessary, further geometry optimization can be performed at any desirable level including *ab initio* quantum chemistry.

Figure 8 shows the overall process of the structure-conversion method explained above. To obtain possible BO matrices from the adjacency matrix of a given molecule, we use information only on atomic valences as input parameters. Then, the energetically most stable BO matrix is selected according to universal criteria which are to maximize the sum of BO matrix and to satisfy total charge conservation. Then, from the selected matrix, we find rings, a main chain, branches, and stereogenic centers using graph traversal algorithms such as Kruskal algorithm, DFS, and BFS, which is used to write the corresponding SMILES string. Subsequently, we convert the SMILES string to a 3D geometry using the OpenBabel program. Finally, this geometry will be further optimized using force field or quantum mechanical methods. This program was coded using Python 2.7[33] with NumPy and SciPy math libraries,[34] and Pybel.[24]

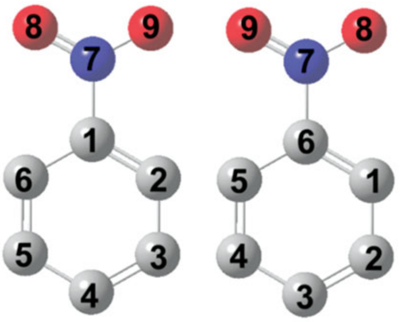


Figure 6. Example of permutational isomers.

Results and Discussion

Accuracy of the Structure Conversion Method. To assess the accuracy of the program we developed, molecular

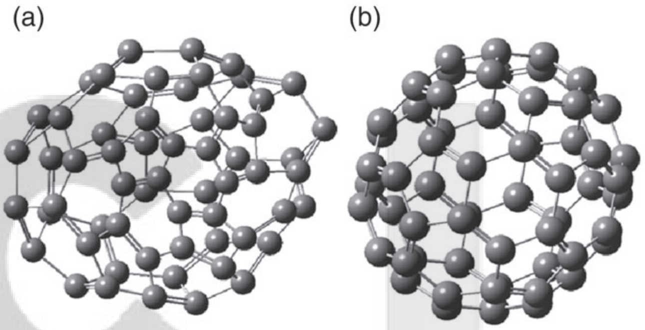


Figure 7. 3D structures of fullerene generated (a) by the OpenBabel program and (b) by the geometry re-optimization using UFF.

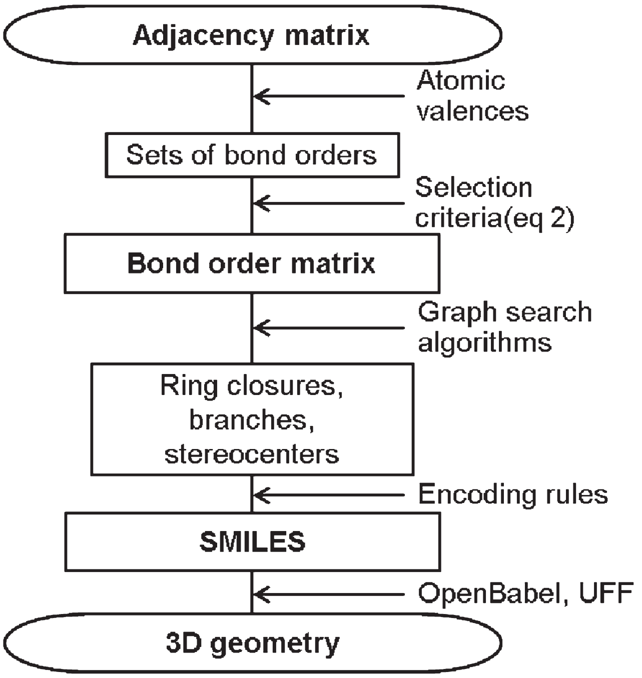


Figure 8. Flow chart of the structure-conversion method.

structures with compound IDs from 1 to 100000 in the PubChem database[27] were sampled in the structure-data file (SDF) format.[35] We randomly chose 10 000 molecules out of samples that have fewer than 100 atoms and that consist of only organic elements (H, B, C, N, O, F, P, S, Cl, and Br). To test our program, we collected information on only the total charges and ACs of the molecules from the SDF files.

First, the accuracy of the BO assignment routine was examined. We made BO matrices from the ACs of the 10 000 samples. The resulting matrices were then compared with those in the original SDF files. In this comparison, we used all possible sets of BOs for each AC, without the selection process of an optimal set of BOs, because BOs in PubChem may be obtained with different criteria from those used in our method. If any of the BO sets was identical to that in PubChem, we regarded it as success. To distinguish permutational isomers due to different numbering of atoms as illustrated in Figure 6, we define another alternative Coulomb matrix $\mathbf{C}^B$ that is constructed from a BO matrix and atomic numbers:

$$C_{ij}^B = \begin{cases} B_{ij} \cdot Z_i \cdot Z_j & \text{if } i \neq j \\ Z_i^2 & \text{otherwise} \end{cases} \qquad (4)$$

Then, we compare between the eigenvalues of the BO matrices from our method and those from PubChem.

Next, we evaluate the accuracy of the 3D geometry conversion from SMILES strings. We compare the 3D coordinates obtained using our method with those in PubChem. This comparison is not straightforward because 3D coordinates are continuous variables, while SMILES contains discrete information. Infinite numbers of 3D coordinates can be built from one SMILES string. Moreover, we do not know how the 3D structures in PubChem were obtained. To make the comparison effective, we introduce the following simple criterion. The 3D structure of a molecule is highly affected by the choice of BO matrix. Especially, bond lengths and angles are key variables in determining molecular structures. These may be reflected in the adjacency matrix that is obtained as

$$A_{ij} = \begin{cases} 1 & \text{if } i \neq j \text{ and } r_{ij} \leq 1.1\left(R_i + R_j\right) \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

where r_{ij} is the interatomic distance between atoms i and j, and R_i is the covalent radius of atom i.[36] We first regenerate the adjacency matrix from the 3D geometry obtained using our method and use it to construct the alternative Coulomb matrix $\mathbf{C}^A$ (Eq. (3)). Then, in order to consider possible permutational isomers, we compare the eigenvalues of this matrix with those from PubChem.

Table 3 shows the results for both BO assignment and 3D geometry conversion. The success rate of the BO assignment is very high (99.97%). Our method failed for only 3 molecules of 10 000. Figure 9 shows the structures of the failed molecules with compound IDs. It should be noted that they have either abnormal atomic valences (CID 62352) or abnormal formal charges (CIDs 73876 and 89412). We can further improve our method by including those abnormal atomic valences. The success rate of the 3D geometry comparison is a bit lower (98.36%), but it is still very high considering the indefinite criterion we introduced. These high success rates guarantee that our method can be used for the structure-conversion of general organic molecules. The compound IDs of the 10 000 molecules used in the test are provided in the Supporting Information.

Application to Search for Molecular Isomers. Finding molecular isomers is an important subject.[37–42] Isomers are composed of the same atoms but have different ACs. Here we propose an extremely powerful and efficient method for finding molecular isomers using the adjacency matrix of a molecule as a useful application of our structure-conversion method. For a molecule composed of N_A atoms, we construct an N_AxN_A zero matrix $\mathbf{A}$ and then change its matrix elements A_{ij} to 1 if both atoms i and j have yet to form chemical bonds, *i.e.*, both the i-th row and the j-th column have only zero elements, as schematically shown in Figure 10(a). This is repeated until all the atoms are connected to form a single molecule. As the adjacency matrix is symmetric, it is sufficient to deal only with its upper triangular matrix. Once we scan all the

Table 3. The success rates of BO matrix and 3D geometry conversion for 10 000 samples.

	Criterion of comparison	Success rate (%)
BO matrix	Eigenvalues of Coulomb matrix from BO ($\mathbf{C}^B$)	99.97
3D geometry	Eigenvalues of Coulomb matrix from adjacency ($\mathbf{C}^A$)	98.36

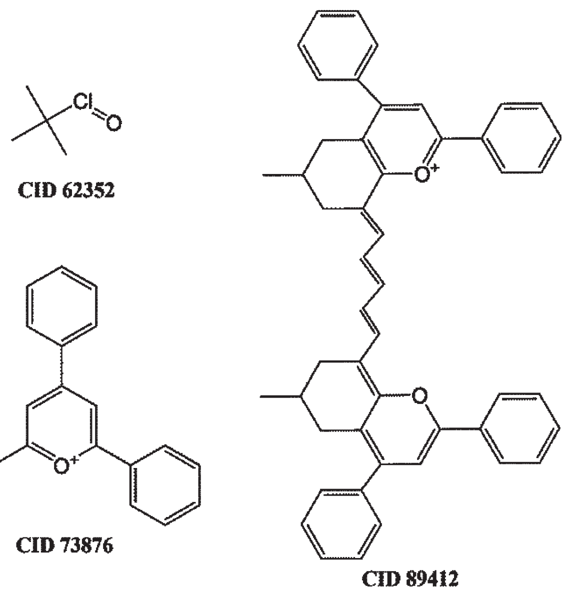


Figure 9. The three molecules that failed in the BO assignment out of 10 000 molecules sampled from the PubChem database and their compound IDs.

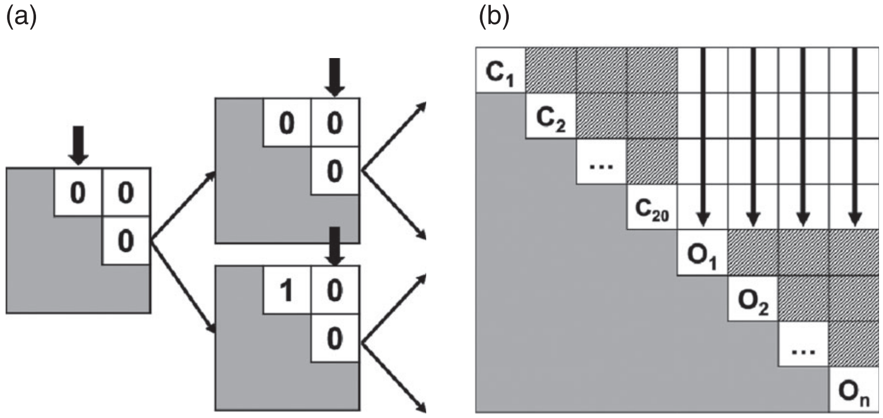


Figure 10. (a) Iterative scheme for the conversion of adjacency matrix elements. The thick arrows indicate the matrix elements where conversion is applied. (b) Constraints on the adjacency matrix of $C_{20}(OH)_n$ to generate isomers using the iterative matrix conversion scheme. New bond formation is allowed not for matrix elements in the shaded area, but for those elements marked by arrows.

possible A_{ij}, this results in the ACs of all the possible isomers. The ACs are converted to corresponding 3D geometries by using our structure-conversion method. We tested this method for two molecules.

As the first example, C_5H_8 was chosen because it is a relatively small molecule that yet has various types of isomers with double/triple bonds, rings, *cis–trans*, and chiral centers. We successfully found all the 33 isomers using our method. Figure 11 shows some of these isomers with SMILES strings, particularly the cyclic/acyclic and cis-trans isomers, and R/S chiral carbons. The SMILES strings and 3D geometries of the remaining isomers are given in the Supporting Information.

As the second example, we chose a more complex system, $C_{20}(OH)_n$. It has as many isomers as there are discrete cases of attachment of n hydroxyl groups on the surface sites of C_{20} fullerene, as depicted in Figure 12. Here, all the possible sets of C─O connectivity were generated using our iterative method (Figure 10). In this case, C_{20} and the O─H bond should remain intact during iteration. Hence, we impose constraints on adjacency matrix elements by allowing bond formation only between C and O, as depicted in Figure 10(b). Table 4 shows the numbers of the isomers of $C_{20}(OH)_n$ for $n \leq 10$; these numbers are consistent with the previous results (for $n \leq 4$) obtained using the genetic algorithm.[43]

Table 4 also shows computational time required to search for molecular isomers. It took 51 s and 50 min to find 58 and 1642 isomers of $C_{20}(OH)_4$ and $C_{20}(OH)_{10}$, respectively, in serial calculations on an Intel® Core™ i5-2500 CPU @ 3.30 GHz processor. It should be noted that most of the time was used for obtaining the 3D geometries via force field calculations. The SMILES strings and 3D geometries of the remaining isomers are shown in the Supporting Information.

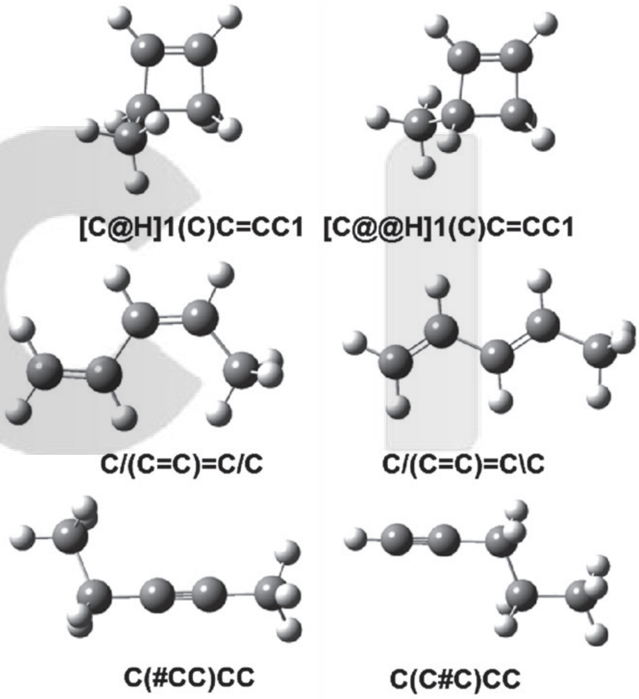


Figure 11. Various isomers of C_5H_8 and their SMILES strings.

Conclusion

We developed a structure-conversion method for organic molecules without resorting to system-dependent parameters. This method enables us to obtain reliable 3D geometries or BO information from just AC. To uniquely determine energetically more favorable structures among all the possible ones, we use universal selection criteria with only atomic valences. As a result, the method can be widely applied to any organic molecules. Indeed, it showed a high success rate in conversions from AC to BOs or 3D geometries for 10 000 molecules sampled from the PubChem database. It should be noted that the slightly lower success rate for 3D geometry conversion is due to the ambiguity in determining the 3D coordinates of molecules. The method's accuracy can be further improved by considering some exceptional cases. As applications, we used our method to find possible isomers for a given chemical

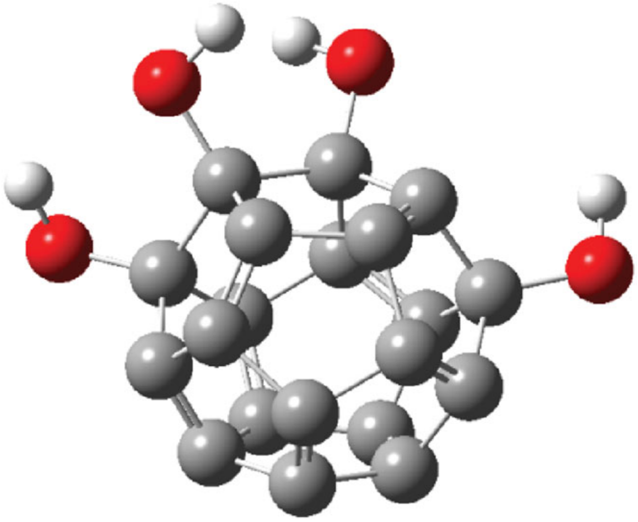


Figure 12. Example of $C_{20}(OH)_4$.

Table 4. The number of isomers found for given chemical formula and corresponding computational times (adjacency matrix generation + geometry generation) on Intel® Core™ i5-2500 CPU @ 3.30 GHz processor.

Chemical formula	Number of isomers	Computational time (adjacency + geometry[a] = total)
C_5H_8	33	1.65 s +18.76 s = 20.41 s
$C_{20}(OH)_1$	1	0.01 s + 0.69 s = 0.70 s
$C_{20}(OH)_2$	5	0.06 s + 3.26 s = 3.32 s
$C_{20}(OH)_3$	15	0.19 s + 11.69 s = 11.88 s
$C_{20}(OH)_4$	58	0.66 s + 50.56 s = 51.22 s
$C_{20}(OH)_5$	149	2.36 s + 2 m 21 s = 2 m 23 s
$C_{20}(OH)_6$	371	8.19 s + 8 m 35 s = 8 m 43 s
$C_{20}(OH)_7$	693	25.61 s + 26 m 32 s = 26 m 59 s
$C_{20}(OH)_8$	1135	1 m 10 s + 33 m 41 s = 34 m 51 s
$C_{20}(OH)_9$	1461	2 m 41 s + 40 m 11 s = 42 m 52 s
$C_{20}(OH)_{10}$	1642	5 m 12 s + 44 m 46 s = 49 m 58 s

[a] Computational time required to optimize geometries using OpenBabel and UFF.

formula. The results show that our method precisely found all the isomers within a very short computational time.

In order to allow for a wider range of applications, we need to deal with metal-containing molecules. However, this may require more careful consideration of atomic valences and formal charges. In addition, the present method is limited to compounds that can be described by SMILES. As a result, it may have problems with radical compounds that SMILES has difficulty handling. Nevertheless, as demonstrated by its powerful isomer searching, our method's simple but thorough scheme allows us to efficiently manipulate the formation and dissociation of chemical bonds. Therefore, we expect that it will be a very useful tool for various chemical applications.

Acknowledgments. This work was supported by the Basic Science Research Program (NRF-2012R1A1A1004154), the EDISON Program (NRF-2012M3C1A6035359), and the MIR center (No.20090083525) through the NRF, funded by the Korean Government (MSIP). W.Y.K. is also grateful for financial support from the Chung-Am Fellowship and the EWon assistant professorship.

Supporting Information. The compound IDs of 10 000 PubChem molecules; and the isomers of C_5H_8 and $C_{20}(OH)_n$ ($1 \le n \le 10$) generated from our method.

References

1. J. Wang, W. Wang, P. A. Kollman, D. A. Case, *J. Mol. Graph. Model.* **2006**, *25*, 247.
2. S. Nikolić, N. Trinajstić, Z. Mihalid, *J. Math. Chem.* **1993**, *12*, 251.
3. A. Ratkiewicz, T. N. Truong, *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 36.
4. A. Ratkiewicz, T. N. Truong, *Int. J. Quantum Chem.* **2006**, *106*, 244.
5. Y. Kim, S. Choi, W. Y. Kim, *J. Chem. Theory Comput.* **2014**, *10*, 2419.
6. M. Hendlich, F. Rippmann, G. Barnickel, *J. Chem. Inf. Model.* **1997**, *37*, 774.
7. Sayle, R. PDB: cruft to content (perception of molecular connectivity from 3D coordinates). In MUG'01 Presentation. Daylight Chemical Information Systems Inc., 2001. URL http://www.daylight.com/meetings/mug01/Sayle/m4xbondage.html.
8. J. C. Baber, E. E. Hodgkin, *J. Chem. Inf. Model.* **1992**, *32*, 401.
9. P. Labute, *J. Chem. Inf. Model.* **2005**, *45*, 215.
10. Q. Zhang, W. Zhang, Y. Li, J. Wang, L. Zhang, T. Hou, *J. Cheminform.* **2012**, *4*, 26.
11. S. V. Trepalin, A. V. Yarkov, I. V. Pletnev, A. A. Gakh, *Molecules* **2006**, *11*, 219.
12. D. Weininger, *J. Chem. Inf. Model.* **1988**, *28*, 31.
13. D. Weininger, A. Weininger, J. L. Weininger, *J. Chem. Inf. Model.* **1989**, *29*, 97.
14. D. Weininger, *J. Chem. Inf. Model.* **1990**, *30*, 237.
15. R. C. Read, *J. Chem. Inf. Model.* **1983**, *23*, 135.
16. R. G. A. Bone, M. A. Firth, R. A. Sykes, *J. Chem. Inf. Model.* **1999**, *39*, 846.
17. C. G. Gotlieb, D. G. Corneil, *Commun. ACM* **1967**, *10*, 780.
18. C. J. Lee, Y. Kang, K. Cho, K. T. No, *Proc. Natl. Acad. Sci. U.S.A.* **2009**, *106*, 17355.
19. J. Gasteiger, C. Jochum, *J. Chem. Inf. Model.* **1979**, *19*, 43.
20. J. B. Kruskal, *Proc. Am. Math. Soc.* **1956**, *7*, 48.
21. J. Kleinberg, É. Tardos, *Algorithm Design*, 1st ed., Boston, MA, Pearson Education, 2006, p. 92.
22. E. Moore, The shortest path through a maze. In Proceedings of the International Symposium on the Theory of Switching, Cambridge, MA, April 2–5, 1957. Harvard University Press, Cambridge, MA, 1959, p. 285.
23. N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, G. R. Hutchison, *J. Cheminform.* **2011**, *3*, 33.
24. N. M. O'Boyle, C. Morley, G. R. Hutchison, *Chem. Cent. J.* **2008**, *2*, 5.

25. Q. Li, T. Cheng, Y. Wang, S. H. Bryant, *Drug Discov. Today* **2010**, *15*, 1052.
26. E. E. Bolton, Y. Wang, P. A. Thiessen, S. H. Bryant, In *Annual Reports in Computational Chemistry*, Vol. 4, A. W. Ralph, C. S. David Eds., Elsevier, Bethesda, MD, 2008, p. 217.
27. Compounds in PubChem database. URL ftp://ftp.ncbi.nlm.nih.gov/pubchem/Compound/CURRENT-Full/SDF/ (accessed July 29, 2014).
28. J. Xu, *J. Chem. Inf. Model.* **1996**, *36*, 25.
29. J. Sadowski, J. Gasteiger, *Chem. Rev.* **1993**, *93*, 2567.
30. A. K. Rappe, C. J. Casewit, K. S. Colwell, W. A. Goddard, W. M. Skiff, *J. Am. Chem. Soc.* **1992**, *114*, 10024.
31. M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery Jr.. , J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, N. J. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, Ö. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, D. J. Fox, *Gaussian 09, Revision A.02*, Gaussian, Inc., Wallingford, CT, 2009.
32. A. K. Rappe, W. A. Goddard, *J. Phys. Chem.* **1991**, *95*, 3358.
33. G. Rossum, *Python Reference Manual*, Centre for Mathematics and Computer Science (CWI), Amsterdam, 1995.
34. T. E. Oliphant, *Comput. Sci. Eng.* **2007**, *9*, 10.
35. A. Dalby, J. G. Nourse, W. D. Hounshell, A. K. I. Gushurst, D. L. Grier, B. A. Leland, J. Laufer, *J. Chem. Inf. Model.* **1992**, *32*, 244.
36. B. Cordero, V. Gómez, A. E. Platero-Prats, M. Revés, J. Echeverría, E. Cremades, F. Barragán, S. Alvarez, *Dalton Trans.* **2008**, 2832.
37. C. Benecke, R. Grund, R. Hohberger, A. Kerber, R. Laue, T. Wieland, *Anal. Chim. Acta* **1995**, *314*, 141.
38. T. Wieland, A. Kerber, R. Laue, *J. Chem. Inf. Model.* **1996**, *36*, 413.
39. I. Lukovits, *J. Chem. Inf. Model.* **1999**, *39*, 563.
40. I. Lukovits, *J. Chem. Inf. Model.* **2000**, *40*, 361.
41. T. Fink, J. Reymond, *J. Chem. Inf. Model.* **2007**, *47*, 342.
42. L. C. Blum, J. Reymond, *J. Am. Chem. Soc.* **2009**, *131*, 8732.
43. M. A. Addicoat, A. J. Page, Z. E. Brain, L. Flack, K. Morokuma, S. Irle, *J. Chem. Theory Comput.* **2012**, *8*, 1841.