

# Automatic atom type and bond type perception in molecular mechanical calculations

Junmei Wang<sup>a,\*</sup>, Wei Wang<sup>b</sup>, Peter A. Kollman<sup>c</sup>, David A. Case<sup>d</sup>

<sup>a</sup> College of Chemistry, Peking University, Beijing 100871, China

<sup>b</sup> Department of Chemistry and Biochemistry, Center for Theoretical Biological Physics,  
University of California at San Diego, La Jolla, CA 92093-0359, United States

<sup>c</sup> Department of Pharmaceutical Chemistry, University of California at San Francisco, San Francisco, CA 94143-0446, United States

<sup>d</sup> Department of Molecular Biology, TPC15, The Scripps Research Institute, 10550 N. Torrey Pines Rd. La Jolla, CA 92037, United States

Received 31 October 2005; received in revised form 16 December 2005; accepted 16 December 2005

Available online 3 February 2006

## Abstract

In molecular mechanics (MM) studies, atom types and/or bond types of molecules are needed to determine prior to energy calculations. We present here an automatic algorithm of perceiving atom types that are defined in a description table, and an automatic algorithm of assigning bond types just based on atomic connectivity. The algorithms have been implemented in a new module of the AMBER packages. This auxiliary module, antechamber (roughly meaning “before AMBER”), can be applied to generate necessary inputs of *leap*—the AMBER program to generate topologies for minimization, molecular dynamics, etc., for most organic molecules. The algorithms behind the manipulations may be useful for other molecular mechanical packages as well as applications that need to designate atom types and bond types.

© 2005 Elsevier Inc. All rights reserved.

**Keywords:** Atom type perception; Bond type perception; Antechamber; Residue topology; Force field parameters; General AMBER force field (GAFF)

## 1. Introduction

Molecular mechanics models are useful in studying structures, conformational energies and other molecular properties, including vibrational frequencies, conformational entropies and dipole moments, etc. [1–21]. Compared to first principle approaches, molecular mechanics models are much faster and simpler owing to the fact that electrons are treated implicitly in MM. When a MM force field is well parameterized, it can achieve an accuracy that is comparable to high-level quantum mechanical models in calculating the relative conformational energies of a molecule [3]. For large systems (proteins, DNA and RNA, etc.), MM is almost the only practical choice given the fact that calculations on large biological systems need massive computer resource and are very time-consuming. MM models also play an important role in rational molecular design especially for drug lead identification using MM-based scoring functions, which needs

to screen a great number of compounds and speed, is therefore a critical factor to be concerned.

To conduct structure-based molecular design, one often needs to study the structures and energies of biological macromolecules (proteins, DNA and RNA) in complex with small drug molecules. In principal, the MM force fields that describe biological molecules and small molecules should be consistent. The traditional AMBER force fields, which were parameterized to study biological macromolecules, have limited parameters for organic molecules. Recently, we have described an organic force field—general AMBER force field (GAFF) [4], which applies the same harmonic energy function as the additive AMBER force fields do:

$$E_{\text{pair}} = \sum_{\text{bonds}} K_r (r - r_{\text{eq}})^2 + \sum_{\text{angles}} K_{\theta} (\theta - \theta_{\text{eq}})^2 + \sum_{\text{dihedrals}} \frac{V_n}{2} \times \left[ 1 + \cos(n\phi - \gamma) \right] + \sum_{i < j} \left[ \frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^6} + \frac{q_i q_j}{\epsilon R_{ij}} \right] \quad (1)$$

Here,  $K_r$ ,  $K_{\theta}$  and  $V_n$  are the force constants for bond length stretching, bond angle bending and torsional angle twisting, respectively;  $r_{\text{eq}}$  and  $\theta_{\text{eq}}$  are the equilibrium bond lengths and

\* Corresponding author.

E-mail addresses: [jwang@encysive.com](mailto:jwang@encysive.com) (J. Wang),  
[case@scripps.edu](mailto:case@scripps.edu) (D.A. Case).

bond angles, respectively;  $\gamma$  is the phase angles of the Fourier series in the dihedral terms;  $A_{ij}$  and  $B_{ij}$  are the parameters of Lennard–Jones 12-6 potentials;  $q_i$  and  $q_j$  are the point charges of atoms  $i$  and  $j$ , respectively. GAFF can describe most organic molecules in a fashion that is reasonably accurate and consistent with the way in which traditional AMBER force fields describe proteins and nucleic acids.

Nevertheless, the development of a general AMBER force field is only a part of the story. To efficiently handle a great number of small molecules in MM calculations, one needs to automatically assign atom types, bond types and then generate proper “topologies” that encode force field parameters ( $K_r$ ,  $K_\theta$ ,  $V_n$ ,  $r_{eq}$ ,  $\theta_{eq}$ ,  $\gamma$ ,  $A_{ij}$ ,  $B_{ij}$ , etc.) for arbitrary organic molecules and their biological complexes. In the following, we present the algorithms and procedures beyond the manipulations.

Considering that the bond type information may be used in atom type definition, we first introduce the bond type perception algorithm followed by the atom type perception algorithm. Finally, the application of these two algorithms in molecular mechanical calculations is presented.

## 2. Bond type perceptions

### 2.1. Introduction

For some reasons, bond type information is usually not applied in most widely used force fields. Why then is a bond type perception algorithm necessary? First of all, as being further explained in the next section, bond types may be needed to precisely define atom types (see Table 4, for examples). Secondly, bond types are needed to generate charges efficiently and automatically for arbitrary molecules with the AM1-bond charge correction (BCC) scheme. In AMBER, ideally, the HF/6-31G\* restrained electrostatic potential (RESP) charge model [22–25] should be applied in MM calculations since the force fields were parameterized with this charge model. However, RESP is an expensive model that needs a HF/6-31G\* geometry optimization; thus, it is not practical to generate charges for a large number of molecules. Recently, Jakalian et al. have developed a fast and efficient charge model, which is called AM1-BCC [26,27]. The AM1-BCC scheme first calculates the Mulliken charges at the AM1 semi-empirical level, and then conducts bond charge corrections to generate the RESP-like charges. The BCC parameters were derived using a large training set to make electrostatic potentials calculated with AM1-BCC point charges reproduce the ab initio (HF/6-31G\*) ones. In most cases, the AM1-BCC charge performs comparable to that of RESP and therefore is a good substitute for RESP in molecular design. As the BCC parameters are both atom and bond type dependent, one needs to recognize both atom types and bond types to generate AM1-BCC charges.

Many popular molecular formats, such as *mdl* and *mol2*, have a bond type field. Unfortunately, the bond types are usually too simple to be used directly. For example, *mdl* has only three bond types: single, double and triple and *mol2* has two more: aromatic and amide. The following reasons justify us to develop a new algorithm to assign bond types. First of all,

many molecular formats do not have bond type information at all, such as Gaussian Cartesian (*gcrt*), Gaussian Z-matrix (*gzmat*), Gaussian output or log file (*gout*), Mopac Cartesian (*mopcrt*), Mopac Z-matrix (*mopint*) and Mopac output or log file (*mopout*). The widely used *pdb* format has only connectivity information for non-standard residues. Secondly, the three atom types in *mdl* and the five atom types in *mol2* may not be sufficient to define atom types concisely and efficiently. Finally, seven bond types are defined for AM1-BCC, which are single, double, triple, aromatic single, aromatic double, dative bond (e.g. N-oxide), delocalized bond (e.g. nitro and carboxyl). Here, we introduce an efficient algorithm to recognize the seven bond types only using the bond connectivity information.

### 2.2. Basic concepts

Several concepts deserve introduction here. An atom's connectivity (*con*) is the number of bonded atoms. For example, the connectivity of a  $sp^3$  carbon is 4 and that of a  $sp^2$  carbon is 3. Connectivity information is read in from the input in either *ac* (a modified *pdb* format) or *mol2* format. The second concept, atomic valence (*av*) is defined by Eq. (2).

$$av = \sum_{i=1}^{con} bo_i \quad (2)$$

$$tps = \sum_{i=1}^n aps_i \quad (3)$$

where  $bo_i$  is bond order of the bond between the atom in question and the  $i$ th bonded atom, which takes a value of 1 (single bond), 2 (double bond) or 3 (triple bond). The third concept is the atomic penalty score (*aps*) of an atom, which is 0 if its *av* has the most desirable value, or an integer if not. For example, for carbon, *aps* is 0 if *av* equals to 4 (the most desirable value), and is 32 if *av* equals to 3 or 5 (less desirable values). The basic concepts and terms introduced in this work for bond type and atom type perception are summarized in Table 1. Table 2 lists the *aps* values for 35 atom types for bond type assignment. It is pointed out that the 35 atom types have nothing to do with atom types defined in a description table to be discussed in the next section. The program can then enumerate all possible valence states for an input structure, and pick out those that have the lowest total penalty scores (*tps*) defined in Eq. (3).

Most molecules have multiple valence states since many atoms have more than one atomic valence. The smaller the *tps*, the more reasonable a structure is. Fig. 1 shows some molecules in different valence states. In this figure, the abnormal atomic valences are labeled near the atoms that cause the abnormality. For each molecule, the valence state on the farthest left, which has the smallest *tps*, is the most reasonable assignment according to common sense. The total number of valence states can be enormous even for a small or middle-sized molecule. Suppose each atom has two atomic valences, then the total

Table 1  
Basic concepts and terms introduced in this work for atom type and bond type perception

Concept/term	Symbol	Description
<b>Bond type perception</b>		
Atomic connectivity	con	Number of bonded atoms
Atomic valence	av	Sum of bond orders of an atom's bonds
Atomic penalty score	aps	Penalty score for an atomic valence, 0 if av is the most desirable value
Total penalty score	tps	Total atomic penalty score for a molecule
<b>Atom type perception</b>		
Atomic path		A collection of atoms (1, 2, 3, ..., $i - 1$ , $i$ , $i + 1$ ) in sequence, where the neighboring atoms (such as $i - 1$ and $i$ ) and/or the first and the last atoms are bonded
Ring type	RG3–RG9	A ring is formed when the first and the last atoms of an atomic path are bonded; RG3–RG9 designate three to nine-membered ring
<b>Aromatic type</b>		
	AR1	Pure aromatic ring
	AR3	Planar ring that has one or several double bonds formed with outside-ring atoms
	AR2	Planar ring other than AR3
	AR5	Pure aliphatic ring
	AR4	Ring other than AR1, AR2, AR3 and AR5
<b>Bond types</b>		
	sb	Single bond
	db	Double bond
	tb	Triple bond
	AB	Aromatic bond
	SB	Single bond including aromatic single
	DB	Single bond including aromatic double
	DL	Delocalized bond
Atomic property string		The ring types, aromatic types and bond types are combined in a string with logical “and” and “or” operations
Chemical environment string		The atomic paths, atomic connectivity and atomic property are combined in a string to describe an atom's environment
Bond connectivity in a chemical environment string		Bonds formed and/or bond type information of two atoms in a chemical environment string

number of valence states of a molecule is  $2^n$ , where  $n$  is the number of atoms. Therefore, it is important for us to screen out the valence states with the lowest tps efficiently.

### 2.3. Bond type assignment procedure

A recursive function, which is the core part of the bond type perception program, *bondtype*, is applied to find all the combinations of valence states for a given tps. This function is repeatedly called for tps equaling to 0, 1, 2, 3, ... until the number of accumulated valence states reaches a predefined limit, which is set to 2000 in default. For a certain valence state, a bond order assigning function (boaf) is applied to determine the bond order for each bond. This function returns 1 if the assignment is successful and 0 if not. Boaf is called iteratively for a set of sorted valence states starting from the ones with the smallest tps until a successful assignment is achieved.

The following is the basic rules that boaf uses to assign bond orders for a given valence state: (1) for each atom in a bond, if

the bond order bo is determined, con is deducted by 1 and av is deducted by bo; (2) for one atom, if its con equals to av, the bond orders of its unassigned bonds are set to 1; (3) for one atom, if its con equals to 1, the bond order of the last bond is set to av; (4) if all the bonds are successfully assigned, con and av of every atom are both 0. If it is the case, boaf returns 1 and stops. Taking buta-1,3-diene (Fig. 2a) as an example, for the most favorable valence state, all the hydrogen atoms have atomic valence of 1 and all the carbons have atomic valence of 4. In the first step, the bond orders of those bonds that involve hydrogen are set to 1 based on Rule (2) or (3); then the bond orders of C1–C2 and C3–C4 are set to 2 based on Rule (3) since C1 and C4 now have connectivity of 1 and atomic valence of 2; finally, the bond order of C2–C3 is set to 1 according to Rule (2) or (3). The atomic connectivity and atomic valence of each atom is adjusted according to Rule (1) after each bond order assignment. For instance, after the first assignment all hydrogen atoms have both con and av of 0; C1 and C2 have con of 1 and av of 2; C2 and C3 have con of 2 and av of 3.

Table 2  
Atomic penalty scores (aps) for different atom types in bond type perception

No.	Atom type	av0	av1	av2	av3	av4	av5	av6	av7
1–5	H, F, Cl, Br, I	64	0	64					
6	C in C≡N–R				0	1	32		
7	C(X1)				1	0	32		
8	C in COO <sup>−</sup>					32	0	32	
9	C			64	32	0	32	64	
10	Si					0			
11	N(X1) in N=N=R			0	0				
12	N(X1)			3	0	32			
13	N(X2) in N=N=R				1	0			
14	N(X2)			4	0	2			
15	N(X3) in nitro				64	32	0	32	
16	N(X3) in pyridine-1-oxide, etc.				1	0			
17	N(X3)			32	0	1	2		
18	N(X4)			64	0	64			
19	O(X1) in pyridine-1-oxide, etc.	0	1						
20	O(X1)	1	0	64					
21	O(X2)	32	0	64					
22	P(X1)			2	0	32			
23	P(X2)			4	0	2			
24	P(X3)			32	0	1	2		
25	P(X4) is bonded to two O(X1) or S(X1)						32	0	32
26	P(X4) is bonded to three O(X1) or S(X1)							32	0
27	P(X4)				64	1	0	32	
28	S(X1) in pyridine-1-thiol anion, etc.	0	1						
29	S(X1)	2	0	64					
30	S(X2)	32	0	32	1				
31	S(X3)				1	0	2	2	
32	S(X4) is bonded to two O(X1) or S(X1)							0	32
33	S(X4) is bonded to three O(X1) or S(X1)							32	0
34	S(X4) is bonded to four O(X1) or S(X1)							32	0
35	S(X4)					4	2	0	

In this table, av0–7 stands for atomic valences 0–7. In the atom type column, X1–X4 denotes that the number of bonded atoms is 1–4.

In some circumstances, the bond order of a bond is difficult to be determined only applying the above rules. In this situation, a trial and error test is carried out as follows: in the first place, the program assumes that the bond order is 1 and pursues bond order assignment according to the basic rules; if discrepancy happens, the bond order is reset to 2 and then 3. Here, discrepancy means av is not 0 when con is, or av is 0 when con is not. If discrepancies happen for all three bond orders, the program boaf exits and returns 0. The procedure of bond order assignment for benzene is shown in Fig. 2b. After the first step assignment, all the carbon atoms have con of 2 and av of 3. Therefore, none of the basic rules can be applied to continue the assignment. The trial and error test assumes the bond order of C1–C2 is 1 and then carries out bond order assignment step by step according to the basic rules.

If boaf fails for all the saved valence states, a warning message is given. After the bond orders are determined, the seven bond types according to the AM1-BCC's definitions can be easily assigned. For example, if the bond order is three, the bond type should be a triple bond; if it is two, the bond is an aromatic double bond in case of the two atoms being aromatic atoms, or a double bond for other cases; if it is one, it is an aromatic single bond for aromatic atoms, or a single bond for

others. According to the AM1-BCC's definitions, the aromatic atoms have AR1 or AR2 aromatic properties (see next section). Finally, delocalized bond type is assigned to C–O bonds in carboxyl anions and nitro groups.

#### 2.4. Limitations

The bond orders in this algorithm are integers (that is, they are formal bond orders) and have different meanings from those calculated by ab initio approaches. Some molecules, such as naphthalene and benzene, have more than one way of assigning alternative aromatic single and double bonds. In this situation, the bond type assignment depends upon the sequence order of atoms in the input. Fortunately, for AM1-BCC, the BCC parameters of aromatic single and double bonds are identical and either assignment gives the correct AM1-BCC charges. It is noteworthy that this algorithm may not work for molecules with unusual valence states, such as radicals using the default atomic penalty scores listed in Table 2. In this situation, the aps parameters may be changed for handling those specific molecules. It is pointed out that the bond type assigning program, *bondtype*, is a stand-alone program that can be integrated into other packages.

### 3. Atom type perception

#### 3.1. Introduction

Force field parameters ( $K_r$ ,  $K_\theta$ ,  $V_n$ ,  $r_{eq}$ ,  $\theta_{eq}$ ,  $\gamma$ ,  $A_{ij}$ ,  $B_{ij}$ , etc.) are developed using model molecules to reproduce experimental and high-level ab initio data. The derived parameters are then expected to transfer to molecules outside the training set [3,28–31]. Force field parameter transferability is therefore the key of a successful general molecular mechanical model, which deals with a variety of molecules (those specific force fields that only model a certain kind of molecules (such as proteins) may not have a transferability problem). In order to achieve a good transferability of a general MM model, atom types should be defined carefully. Atoms that have similar electronic, chemical and structural properties should share a same atom type. Different force fields have different strategies to define atom types: some MM software packages offer an in-core engine to

perceive atom types and users cannot change the definitions at will; the others, such as AMBER, read in molecular topologies that encode atom type information. For MM packages like AMBER, it is necessary to work out an algorithm to recognize atom types automatically so that residue topologies can be easily generated for any arbitrary organic molecules. In our algorithm, atom types are defined in a table so that users can add, delete and modify entries easily. The core atom type recognition engine, which checks the definition table entry by entry for the match of the chemical environment of an atom, is invisible to users.

There are a number of atom type description languages available today, which include Simplified Molecular Input Line Entry Specification (SMILES) [32], Sybyl Line Notation (SLN) [33] and Atom Type Description Language (ATDL) implemented in VEGA (a molecular file format conversion program) [34]. SMILES and SLN encode the topological information of a molecule in a string and have their main applications in database searches. On the other hand, ATDL describes local

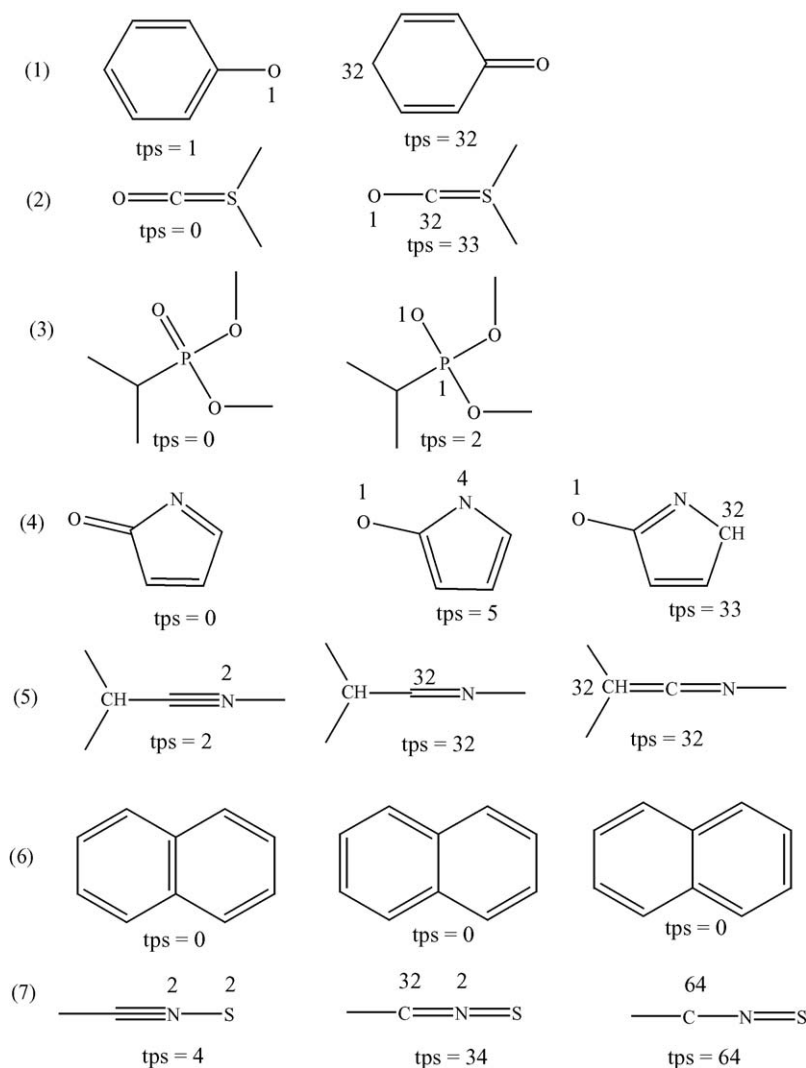


Fig. 1. Exemplary molecules that illustrate how to rank bond order assignments according to their total penalty scores (tps). Abnormal atomic valences are marked near those atoms that cause the abnormality. For each molecule, several possible bond order assignments are shown with one or several tps. The farthest left assignment is the most reasonable according to common sense. All three assignments of No. 6 and the first two assignments of No. 11 have the same penalty scores, and their bond type assignments are affected by the atomic sequence orders in the input.

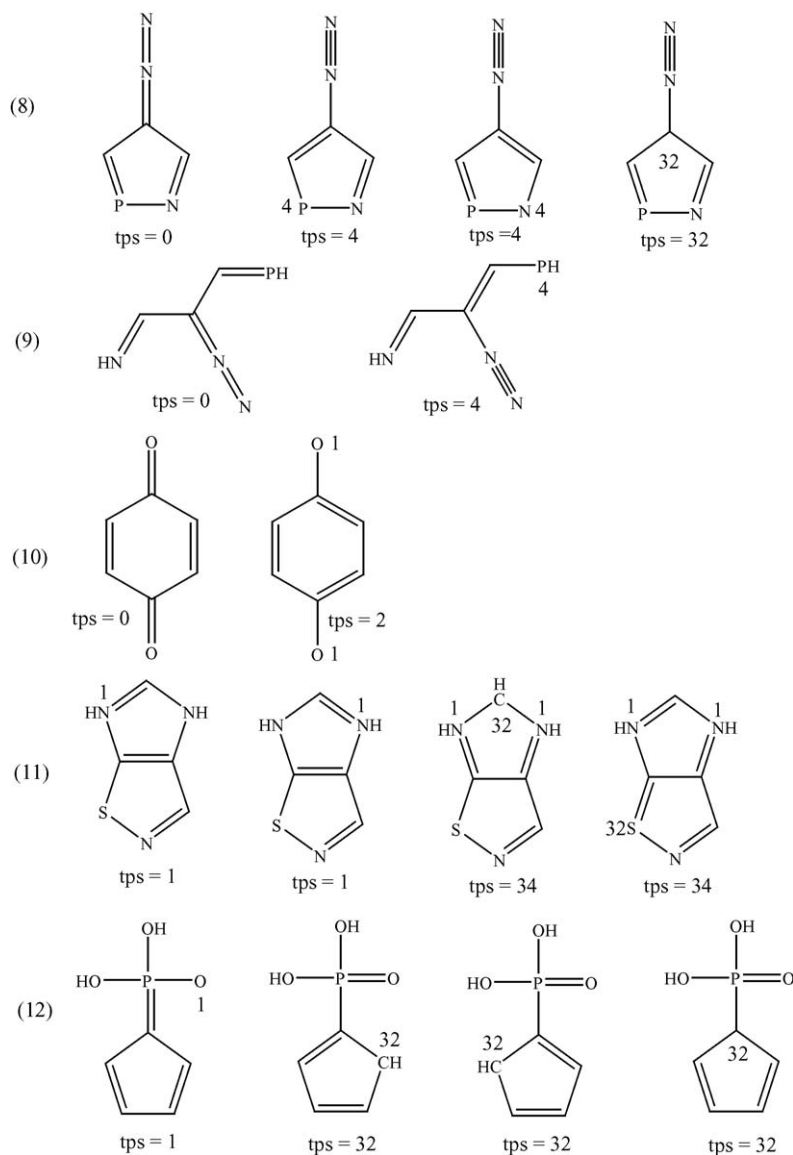


Fig. 1. (Continued).

chemical environments with connectivity, hybridization, etc., and is suitable to assign force field atom types.

Similar to ATDL, we classify atom types mainly based on an atom's electronic and topological properties, which include atomic hybridization, electron donor and acceptor capability, bond types, ring types, aromatic types and the local chemical environment. Specifically, we apply the following features/fields to define atom types: (1) atomic number; (2) number of bonded atoms; (3) number of attached hydrogen atoms; (4) for a hydrogen, number of electron-withdrawing (EW) atoms connected to the atom that hydrogen is bonded to; (5) atomic property (ring type, aromatic type and bond type); (6) precise chemical environments (defined below); (7) bond connectivity in precise chemical environments. The last feature, which is not available in ATDL, provides us additional flexibility in discriminating more subtle chemical environments. We believe that most atom types defined in the widely used force fields (MM3 [5–7], MMFF [16–21], AMBER [1–4], CHARMM [10], OPLS [8], etc.) can be precisely defined with the above

features/fields. Take AMBER force field as an example, the text-based definitions of atom types in the force field files can be understood by a chemist (although ambiguities still present), but not a computer program. Therefore, the above atom type definition feature/fields should be arranged in a way that a computer program can parse easily. Table 3 lists the atom type descriptions for AMBER force fields [1–3]. Regarding the seven features/fields used in the definition table, the first four features are self-explanatory and the last three will be elucidated in detail in the following paragraphs. Tables 4–6 list the atom type descriptions of sulfur in GAFF [4], carbon in MM3 [5–7] and nitrogen in MMFF94 [16–21], respectively. We want to emphasize that the description table is independent of the core engine—the atom type matching program.

### 3.2. Ring system classification and aromatic type definition

The concept of atomic path is extensively used in this work. An atomic path is a collection of atoms (1, 2, 3, ...,  $i - 1$ ,  $i$ ,



Table 3  
The definitions of atom types in the AMBER force field

No.	Type	F1	F2	F3	F4	F5	F6	F7
1	CT	6	4	&				
2	C	6	3	*	*	*	(XA1)	&
3	CN	6	3	*	*	[RG5, RG6, AR1.AR2.AR3]	(C3, C3, N3(H))	&
4	CB	6	3	*	&	[RG5, RG6, AR1.AR2.AR3]	&	
5	CR	6	3	*	*	[RG5, AR1.AR2.AR3]	(N3, N3)	&
6	CR	6	3	*	*	[RG5, AR1.AR2.AR3]	(N2, N3(H))	&
7	CK	6	3	*	*	[RG5, AR1.AR2.AR3]	(N2, N3)	&
8	CC	6	3	0	*	[RG5, AR1.AR2.AR3]	(C3, N3(C3, H))	&
9	CC	6	3	0	*	[RG5, AR1.AR2.AR3]	(C3, N2(C3))	&
10	CW	6	3	*	*	[RG5, AR1.AR2.AR3]	(C3, N3(H))	&
11	CV	6	3	*	*	[RG5, AR1.AR2.AR3]	(C3, N2)	&
12	C*	6	3	*	*	[RG5, AR1.AR2.AR3]	(C3, C3)	&
13	CQ	6	3	*	*	[RG6, AR1.AR2.AR3]	(N2, N2)	&
14	CM	6	3	*	*	[RG6, AR1.AR2.AR3]	(C3, C3(N2(C3(N3(C3))))))	&
15	CM	6	3	*	*	[RG6, AR1.AR2.AR3]	(C3, C3(N3(C3(N3(C3))))))	&
16	CM	6	3	*	*	[RG6, AR1.AR2.AR3]	(C3, N3(C3(N2(C3(C3))))))	&
17	CM	6	3	*	*	[RG6, AR1.AR2.AR3]	(N3, C3(C3(N3(C3(N3))))))	&
18	CA	6	3	*	&	[AR1.AR2.AR3]	&	
19	CD	6	3	*	*	*	(C3, C3)	&
20	CM	6	3	&				
21	CZ	6	2	&				
22	H	1	1	*	*	*	(N)	&
23	HO	1	1	*	*	*	(O)	&
24	HS	1	1	*	*	*	(S)	&
25	HP	1	1	*	*	*	(C(N4))	&
26	HW	1	1	*	*	*	(O(H1))	&
27	H1	1	1	*	1	*	(C4)	&
28	H2	1	1	*	2	*	(C4)	&
29	H3	1	1	*	3	*	(C4)	&
30	HC	1	1	*	*	*	(C4)	&
31	H4	1	1	*	1	*	(XX[AR1.AR2.AR3])	&
32	H5	1	1	*	2	*	(XX[AR1.AR2.AR3])	&
33	HA	1	1	*	*	*	(XX[AR1.AR2.AR3])	&
34	F	9	1	&				
35	Cl	17	1	&				
36	Br	35	1	&				
37	I	53	1	&				
38	N1	7	1	&				
39	NB	7	2	*	*	[RG5, AR1.AR2.AR3]	&	
40	NC	7	2	*	*	[RG6, AR1.AR2.AR3]	&	
41	N2	7	*	*	*	*	(C3(N3, N3))	&
42	NO	7	*	*	*	*	(O1, O1)	&
43	NA	7	3	1	*	[RG5, RG6, AR1.AR2.AR3]	&	
44	N2	7	3	*	*	[NR]	(XX[AR1.AR2.AR3])	&
45	N*	7	3	*	*	[AR1.AR2.AR3]		
46	N	7	3	*	*	*	(C3(XA1))	&
47	NT	7	3	&				
48	N3	7	4	&				
49	O2	8	1	*	*	*	(C(O1))	&
50	O2	8	1	*	*	*	(P)	&
51	O	8	1	*	*	*	(C)	&
52	O	8	1	*	*	*	(S)	&
53	OH	8	2	1	&			
54	OW	8	2	2	&			
55	OS	8	2	&				
56	SH	16	2	1	&			
57	SH	16	2	2	&			
58	S	16	2	&				
59	SO	16	4	&				
60	LP	0	&					
61	DU	&						

F1–F7 is electronic/structural features applied to define atom types. F1: atomic number; F2: number of atoms connected; F3: number of attached hydrogen atoms; F4: number of the electron-withdrawal atoms (N, O, F, Cl and Br) binding to the immediately connected atom; F5: atomic property; F6: chemical environment definition; F7: bond connectivity in chemical environment defined in F6. Some wildcard elements are defined as the following: XX stands for C, N, O, S and P; XA stands for O and S; XB stands for N and P; XD stands for S and P. Seven bond types applied in the atom type definition are listed as the following: sb (single bond), db (double bond), tb or TB (triple bond), AB (aromatic bond), DL (delocalized bond), SB (single bond, including aromatic single and delocalized bond) and DB (double bond, including aromatic double).

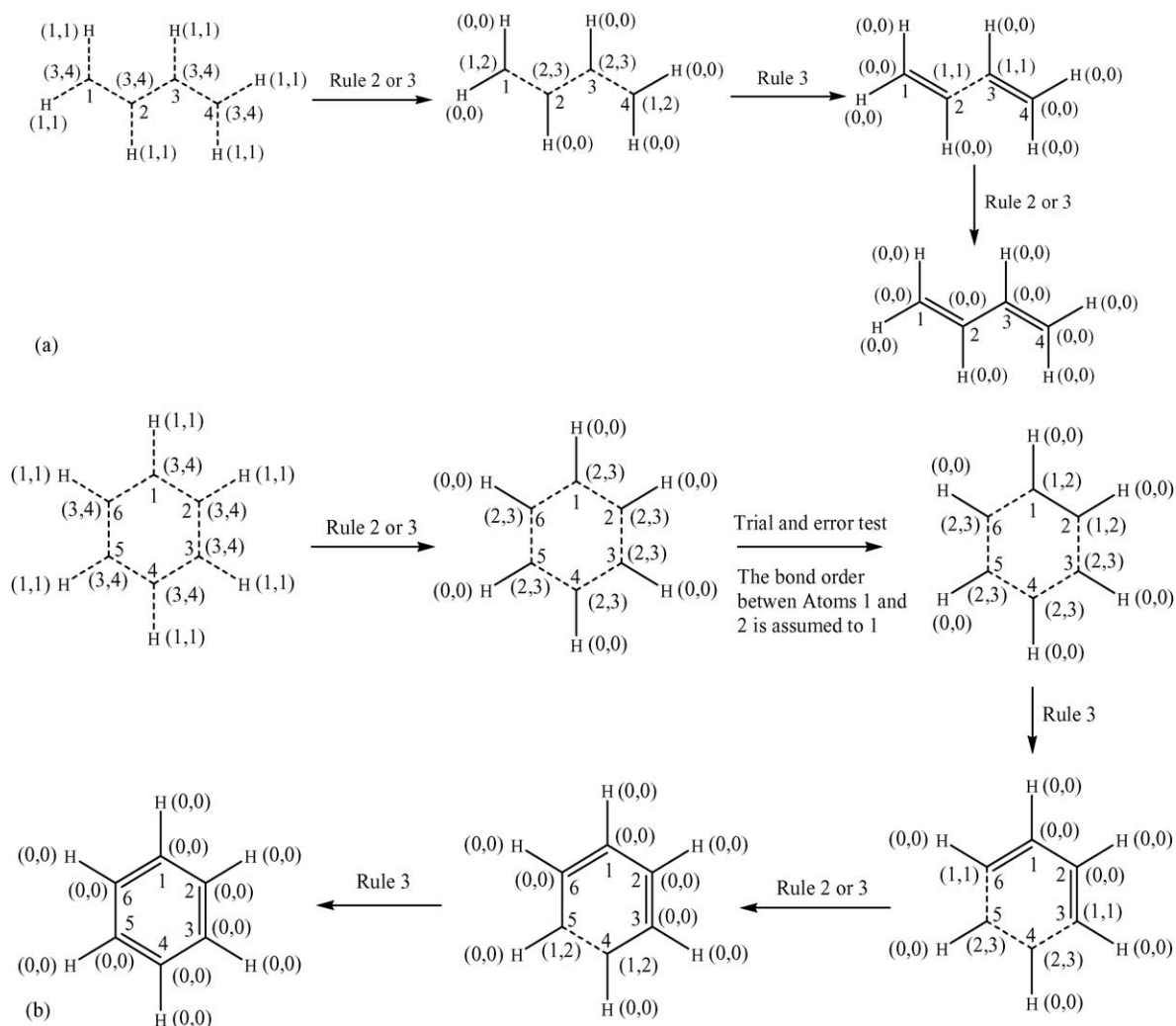


Fig. 2. Examples that demonstrate the procedure of bond order assignment for a given valence state by boaf. con (atomic connectivity) and av (atomic valence) of each atom at each stage of assignment are labeled in a form of (con, av). The dashed bond means its bond order is not determined. The specific rule that is applied at each stage is designated along with the arrow sign. (a) Buta-1,3-diene and (b) benzene.

Table 4  
Definitions of eight sulfur atom types in the general amber force field (GAFF)

No.	Type	F1	F2	F3	F4	F5	F6	F7
1	s	16	1	&				
2	s2	16	2	*	*	[DB]	&	
3	s2	16	2	*	*	[TB]	&	
4	sh	16	2	1	&			
5	sh	16	2	2	&			
6	ss	16	2	&				
7	sx	16	3	*	*	[db]	(XB2{a1})	sa:a1:sb
8	sx	16	3	*	*	[db]	(C3{a1})	sa:a1:sb
9	sx	16	3	*	*	[db]	(XD3[sb,db]{a1})	sa:a1:sb
10	sx	16	3	*	*	[db]	(XD4[sb,db]{a1})	sa:a1:sb
11	s4	16	3	&				
12	sy	16	4	*	*	[db]	(XB2[sb]{a1})	sa:a1:sb
13	sy	16	4	*	*	[db]	(C3[sb]{a1})	sa:a1:sb
14	sy	16	4	*	*	[db]	(XD3[sb,db]{a1})	sa:a1:sb
15	sy	16	4	*	*	[db]	(XD4[sb,db]{a1})	sa:a1:sb
16	s6	16	4	&				
17	s6	16	5	&				
18	s6	16	6	&				

F1–F7 and wildcard elements are defined in Table 3.



Table 5  
Definitions of 14 carbon atom types in MM3<sup>a</sup>

No.	Type	F1	F2	F3	F4	F5	F6	F7
1	22	6	4	*	*	[RG3]	&	
2	56	6	4	*	*	[RG4]	&	
3	1	6	4	&				
4	67	6	3	*	*	[RG3]	(O1)	&
5	58	6	3	*	*	[RG4]	(O1)	&
6	3	6	3	*	*	*	(O1)	&
7	38	6	3	*		[RG3]	&	
8	57	6	3	*	4	[RG4]	&	
9	50	6	3	*	*	[AR1.AR2.AR3]	&	
10	2	6	3	&				
11	68	6	2	*	*	[DB,DB]	(C3,C3)	&
12	106	6	2	*	*	[DB,DB]	(C3,O1)	&
13	71	6	2	*	*	[DB,SB]	(C3,O1)	&
14	4	6	2	&				

F1–F7 and wildcard elements are defined in Table 3.

<sup>a</sup> Atom types 29 (carbon radical), 30 (carbonium ion), 113 (carbon in ferrocene-H) and 114 (carbon in ferrocene-C) are excluded.

Table 6  
Definitions of 24 nitrogen atom types in MMFF

No.	Type	F1	F2	F3	F4	F5	F6	F7
1	38	7	2	*	*	[RG6,AR1.AR2.AR3]	&	
2	39	7	2	*	*	[RG5,AR1.AR2.AR3]	(N)	&
3	65	7	2	*	*	[RG5,AR1.AR2.AR3]	(N)	&
4	65	7	2	*	*	[RG5,AR1.AR2.AR3]	(O)	&
5	65	7	2	*	*	[RG5,AR1.AR2.AR3]	(S)	&
6	66	7	2	*	*	[RG5,AR1.AR2.AR3]	(C3(N))	&
7	66	7	2	*	*	[RG5,AR1.AR2.AR3]	(C3(O))	&
8	66	7	2	*	*	[RG5,AR1.AR2.AR3]	(C3(S))	&
9	79	7	2	*	*	[RG5,AR1.AR2.AR3]	&	
10	53	7	2	*	*	*	(C3,N2)	&
11	53	7	2	*	*	*	(N2,N2)	&
12	9	7	2	*	*	*	(N2)	&
13	46	7	2	*	*	*	(O1)	&
14	48	7	2	*	*	*	(S3)	&
15	48	7	2	*	*	*	(S4)	&
16	9	7	2	&				
17	82	7	3	*	*	[RG5,AR1.AR2.AR3]	(O1)	&
18	69	7	3	*	*	[RG6,AR1.AR2.AR3]	(O1)	&
19	58	7	3	*	*	[RG6,AR1.AR2.AR3]	&	
20	81	7	3	*	*	[RG5,AR1.AR2.AR3]	&	
21	45	7	3	*	*	*	(O1,O1)	&
22	45	7	3	*	*	*	(O1,O1,O1)	&
23	56	7	*	*	*	*	(C3(N3,N3))	&
24	56	7	*	*	*	*	(C3(N2,N3))	&
25	10	7	3	*	*	*	(C3(O1))	&
26	10	7	3	*	*	*	(C3(S1))	&
27	10	7	3	*	*	*	(C3(N2))	&
28	10	7	3	*	*	*	(N2(N2))	&
29	54	7	3	*	*	*	(N2)	&
30	55	7	3	*	*	*	(C3(N1))	&
31	40	7	3	*	*	[AR1.AR2.AR3]	&	
32	40	7	3	*	*	*	(C3)	&
33	40	7	3	*	*	*	(C2)	&
34	67	7	3	*	*	*	(O1)	&
35	8	7	3	&				
36	68	7	4	*	*	*	(O1)	&
37	47	7	1	&				

F1–F7 and wildcard elements are defined in Table 3.

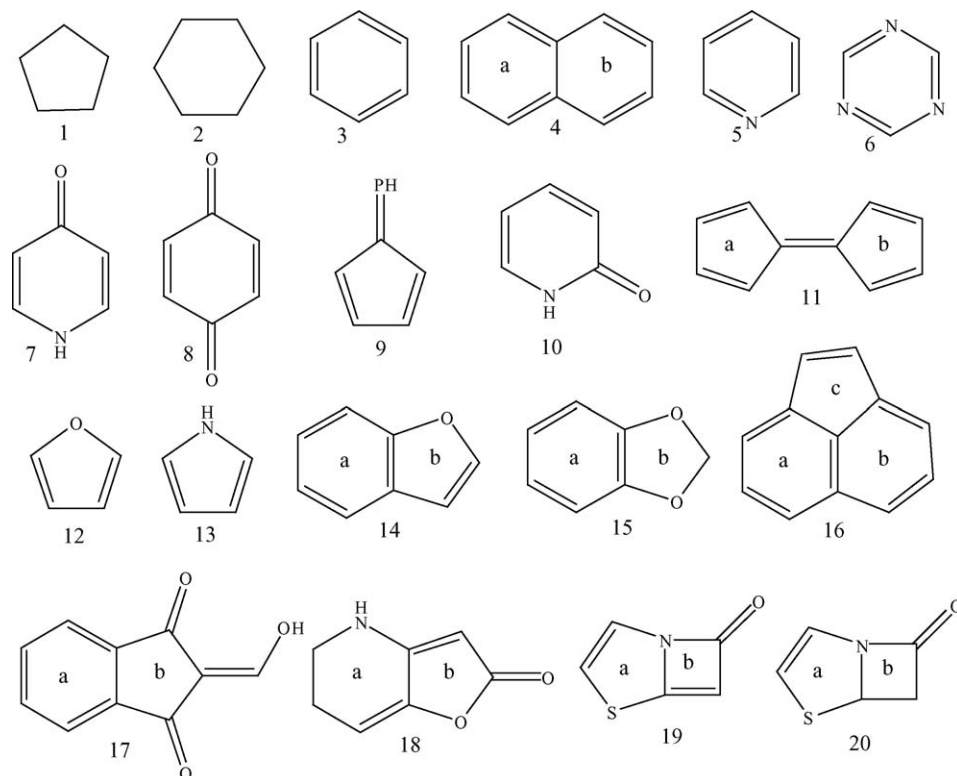


Fig. 3. Exemplary molecules that elucidate the definitions of aromatic types in atom type perception algorithm. Unlabeled atoms are carbon. Ring atoms in molecules 3, 4a, 4b, 5, 6, 14a, 15a, 16a, 16b and 17a are AR1; 1 and 2 are AR5; 7, 8, 9, 10, 17b, 18b and 19b are AR3; 11, 12, 13, 14b and 16c are AR2; others are AR4.

$i + 1, \dots, n$ ) in sequence, where the neighboring atoms (i.e.  $i - 1$  and  $i, i$  and  $i + 1$ ) and/or the head and the tail atoms are bonded. No atom appears more than once in an atomic path. A recursive function is applied to enumerate all the atomic paths. The criterion of forming a ring is atomic path closing, in another word, the head and tail atoms are bonded. In this algorithm, only small (three-membered to nine-membered) rings are counted. The following symbols are used to describe the ring types of an atom: RG (in any ring); RG3–RG9 (in a three-membered to nine-membered ring); NG (not in a ring or in a 10-membered or larger ring).

Five aromatic types are defined for ring atoms in the atom type perception algorithm, which are AR1, AR2, AR3, AR4 and AR5. AR1 denotes atoms in pure aromatic rings, such as the heavy atoms in benzene and pyridine; AR5 designates atoms in pure aliphatic rings, such as the carbons in cyclopentane and cyclohexane; AR3 specifies atoms in planar rings that have one or several double bonds formed with outside-ring atoms, such as carbons in benzoquinone; AR2 is planar ring atoms other than AR3, such as heavy atoms in furan; finally AR4 is ring atoms other than AR1, AR2, AR3 and AR5, such as carbons in cyclopentene and cyclohexene. Here a “planar ring” is defined as a ring that is composed of C(X3), N(X2), N(X3), O(X2), S(X2), P(X2), P(X3), where X2 and X3 denote two and three bonded atoms, respectively. A planar ring usually belongs to one of the three categories: pure aromatic ring (AR1), those forming one or several outside double bonds

(AR3) or those having two continuous single bonds (AR2). Fig. 3 shows some sample molecules that demonstrate how ring types are defined. In this figure, ring atoms in molecules 3, 4a, 4b, 5, 6, 14a, 15a, 16a, 16b and 17a are AR1; 1 and 2 are AR5; 7, 8, 9, 10, 17b, 18b and 19b are AR3; 11, 12, 13, 14b and 16c are AR2; others are AR4.

Besides these ring and aromatic types, bond types are also applied in molecular property definitions (3.3) and chemical environment definitions (3.4). Seven bond types have been defined in this algorithm, which are single bond (sb), double bond (db), triple bond (tb), aromatic bond (AB), single bond including aromatic single and delocalized bonds (SB), double bond including aromatic double (DB) and delocalized bond (DL). It is notable that the bond types used for atom type definition are slightly different from those for AM1-BCC.

### 3.3. Atomic property string

To define an atom type in a convenient and flexible way, atomic property string is introduced. An atomic property string, which is placed in a pair of square brackets, combines the ring property, aromatic property and bond types of an atom with logical “and” and logical “or” operations. For example, [NR] stands for a non-ring atom; [RG5, RG6, AR1.AR2.AR3] defines a five- or six-membered ring atom that has an aromatic type of either AR1 or AR2 or AR3. Here ‘.’ imposes an “or” operation and “,” an “and” operation.

### 3.4. Chemical environment string

To describe an atom's chemical environment precisely, the atomic paths, atomic connectivity and atomic property are collected in a string. The basic unit in a chemical environment string is generic atom that takes a form of “ $XXdd[ap]\langle an \rangle$ ”, where “ $XX$ ” is the name of an element; “ $dd$ ” is the number of bonded atoms of “ $XX$ ”; “ $ap$ ” is the atomic property string of “ $XX$ ”; “ $an$ ” is the generic atom name of “ $XX$ ”. Note that “ $an$ ” is not the really atom name read from an input file, but a generic symbol to discriminate atoms in chemical environment strings. A generic name, which should be unique in a chemical environment string, is needed when one wants to specify a bond formation between this generic atom and another generic atom in the same chemical environment string in the next field. Since the atom for which atom type is being assigned is not a part of chemical environment string, “ $sa$ ” is reserved as the generic symbol of this atom. The symbol “ $sa$ ” is needed when one wants to specify bond formation between the atom in question and an atom in the chemical environment string (see an example below). The following is some examples of generic atoms: “ $C4$ ” defines a carbon with four connected atoms; “ $C4[NR]$ ” defines a non-ring  $sp^3$  carbon; “ $C3[AR1.A-R2.AR3]$ ” defines an  $sp^2$  carbon in a planar ring; “ $N\langle n1 \rangle$ ” defines a nitrogen with any number of connected atoms and having a generic name of “ $n1$ ”.

How are the generic atoms organized in a chemical environment string? The following is the rules: (1) each generic atom belongs to a certain layer in the chemical environment string; (2) each layer is enclosed in a pair of parentheses; (3) atoms in the same layer are separated by “,”; (4) a generic atom  $X$  in the  $i$ th layer is bonded to atoms in the subsequent  $(i + 1)$ th layer when there are no other generic atoms in the  $i$ th layer that separate  $X$  from atoms in the  $(i + 1)$ th layer; (5) for the atom in question, “ $sa$ ” is bonded to atoms in the first layer. In the following, examples are used to explain the above rules. “ $(C,C(O))$ ” defines two layers since there are two pairs of parentheses. The two carbons belong to the first layer since there is only one left parenthesis counted from the left and the oxygen belongs to the second layer since there are two left parentheses counted from the left. The atom in question ( $sa$ ) is bonded with two carbons (Rule 5) and the second carbon is bonded to oxygen (Rule 4). The first carbon is not bonded to the oxygen since it is separated from the oxygen by the second carbon (Rule 4); “ $(C(OI))$ ” means the atom in question ( $sa$ ) is connected with a carbon and this carbon is connected to an oxygen that has only one connected atom; “ $(C3(N3,N3))$ ” means the atom in question ( $sa$ ) is connected with an  $sp^2$  carbon (the first layer), which is connected with two nitrogen atoms (the second layer) that both have three bonded atoms (including the carbon in the first layer). Theoretically, there is no limitation on the number of layers in a chemical environment string.

To further elucidate these rules, a few atom type definition strings from Tables 3–6 are explained in detail. (1) In row 6 of Table 3, the chemical environment string, “ $(N2,N3(H))$ ”, defines two atomic paths: the atom in question being connected to a nitrogen with two bonded atoms; and the atom in question

being connected to another nitrogen with three bonded atoms and one of them is hydrogen. (2) In row 8 of Table 4, the chemical environment string “ $(C3\langle a1 \rangle)$ ” and the bond connectivity in chemical environment field “ $sa:a1:sb$ ”, imply that the atom in question with a default generic name of  $sa$  is bonded to an  $sp^3$  carbon ( $a1$ ) with a bond type of “ $sb$ ”. (3) In row 12 of Table 5, the chemical environment string “ $(C3,O1)$ ” defines two atomic paths: the atom in question being connected to a carbon with three bonded atoms; and the atom in question being connected to a terminal oxygen. (4) In row 23 of Table 6, the chemical environment string “ $(C3(N3,N3))$ ” defines two identical atomic paths: the atom in question being connected to a carbon with three bonded atoms and this carbon is connected to two nitrogen atoms that have three bonded atoms.

With the chemical environment and bond connectivity in the chemical environment fields, one can discriminate subtle local chemical environments and define much more complicated atom types than those listed in Tables 3–6. Three such more complicated definitions for three molecules are shown in Fig. 4. (1) The methyl carbon in acetic acid can be defined as “ $(C3(OI[db],O2(H)))$ ”, here “ $db$ ” means “ $C3$ ”, and “ $O1$ ” forms a double bond. (2) The cyclopentane carbons can be defined as “ $(C4\langle c1 \rangle)(C4(C4\langle c4 \rangle))$ ” with a bond connectivity field of “ $sa:c4:sb$ ” or “ $(C4(C4\langle c2 \rangle),C4(C4\langle c3 \rangle))$ ” with a bond connectivity field of “ $c2:c3:sb$ ”. (3) The chemical environment of the methyl carbon in the third molecule can be defined as “ $(C3(OI,C4\langle c1 \rangle)(C4(C4\langle c3 \rangle)(C4(C4\langle c6 \rangle))))(C4\langle c1 \rangle(C4(C4\langle c3 \rangle)(C4\langle c7 \rangle)))$ ”, where generic atom “ $c6$ ” and atom “ $c7$ ” form a single bond (“ $c6:c7:sa$ ”). The above chemical environment string defines three atomic paths, namely,  $C3-O1$ ,  $C3-C4\langle c1 \rangle-C4\langle c2 \rangle-C4\langle c3 \rangle-C4\langle c5 \rangle-C4\langle c6 \rangle$  and  $C3-C4\langle c1 \rangle-C4\langle c4 \rangle-C4\langle c3 \rangle-C4\langle c7 \rangle$ . The generic atom,  $c3$ , which splices the last two paths, should be named explicitly, although it is not used in the following bond connectivity in the chemical environment field. It is obvious that there are a number of alternative ways to define a same atom type in Field 6 (subtle chemical environment) and Field 7 (bond connectivity in the subtle chemical environment).

Similar to the ring detection scheme, a recursive function is applied in chemical environment matching. All possible atomic paths (not necessarily ending up with terminal atoms) that start from the atom in question are first enumerated. Then the match between the chemical environment of the atom in question and that defined in Fields 6 and 7 of a definition string is checked. A successful match implies that the atomic paths and bond connectivity (defined in the last field) as well as atom properties are totally identical.

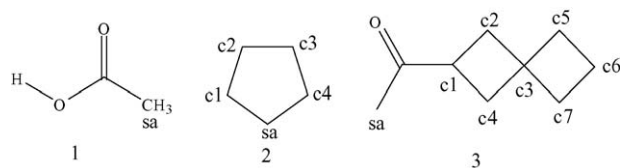


Fig. 4. Exemplary molecules that demonstrate how to define complicated chemical environment strings. “ $sa$ ” is the atom for which an atom type is being assigned and  $c1$ – $c7$  are generic atom names in chemical environment strings.

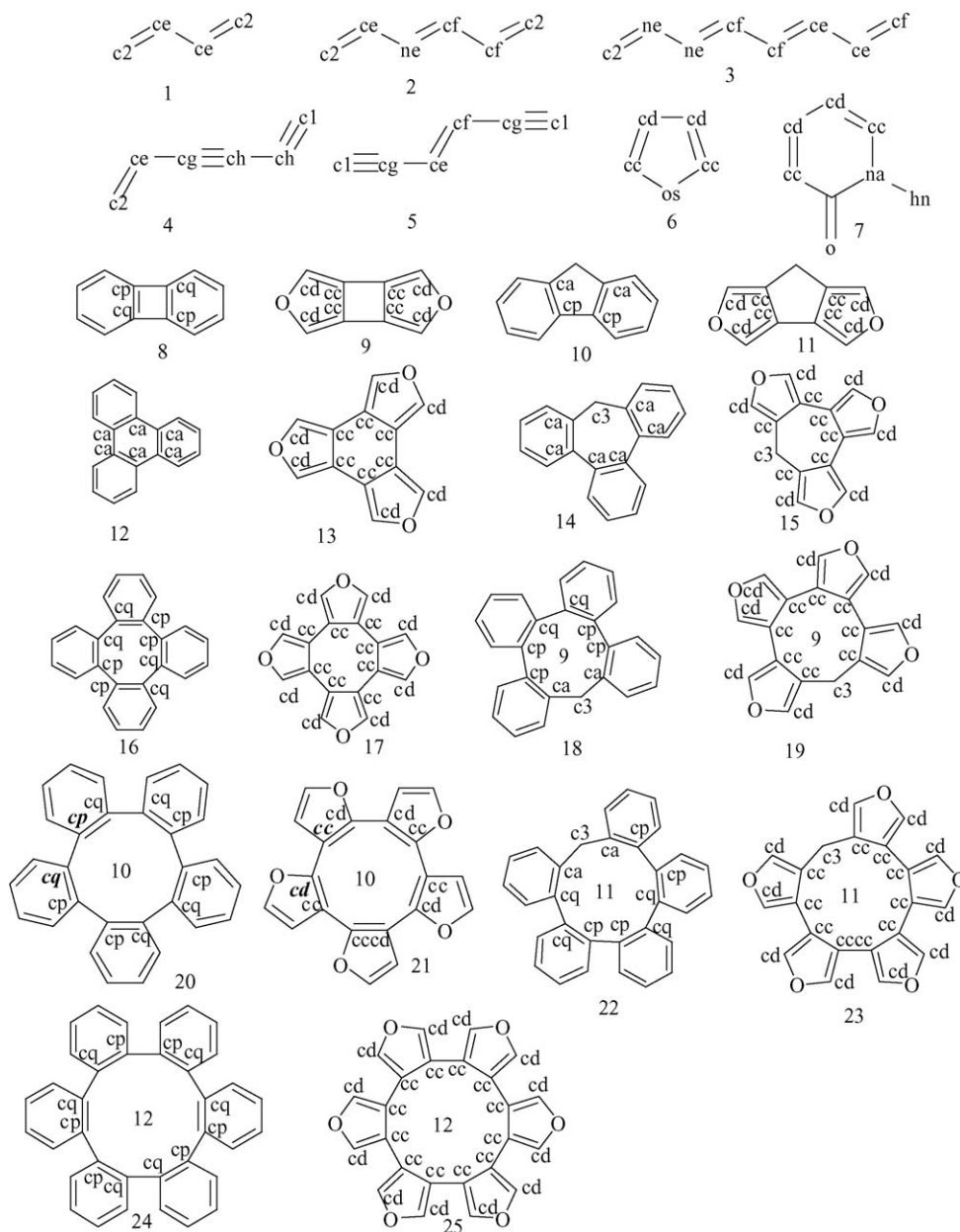


Fig. 5. Exemplary molecules that elucidate the rules of the application of specific atom types to represent bond types. For molecules 20 and 21, the above strategy can not assign atom types correctly and atom types where discrepancy occurs are marked in bold italic.

### 3.5. Atom type assignment procedure

The atom type assignment procedure is simple and straightforward. For an atom type definition string, all the valid fields are checked in sequence. Here a valid field implies that the field is not ‘\*’ or ‘&’, which denotes to neglect the current field and the end of the definition, respectively.

For each atom, a match is performed for each definition string from the beginning. Once a successful match is achieved, the atom type is assigned, and the program jumps from the loop and is ready to handle another atom or halts. Obviously, the sequence order of atom type definitions is crucial for a correct assignment. A rule of thumb is that specific definitions should be placed before the more general ones. If no match is found,

the atom type is finally set to a dummy atom “DU”. Once one or multiple “DU” atom types are assigned, one needs to check the atom type definitions to eliminate the flaws.

It is often believed that bond types (single, double, triple, aromatic single, aromatic double, conjugated single and conjugated double) are essential to precisely describe a chemical environment. However, for some reasons most of biological molecule force fields, including AMBER [1–4], only apply atom types in the force field parameter files. The bond type information is also sacrificed in GAFF [4] for consistency with the AMBER force fields and the programs in the AMBER packages. In order to compensate the lack of bond types in GAFF, several special atom type pairs (*celcf*, *cclcd*, *nelnf*, *nclnd*, *pelpf*, *pclpd*, *cg/ch* and *cp/cq*) were introduced to

discriminate single and double bonds in aromatic and conjugated systems. The first atom type (*cc*, *ne*, *nc*, *pe*, *pc* and *cp*) in each pair is called Group I atom type, while the second atom type (*cf*, *cd*, *nf*, *nd*, *pf*, *pd* and *cq*) in each pair is Group II atom type. How are bond types implicitly represented by the special atom types? The following is the basic rules: for an inner bond A–B in a conjugated/aromatic system, if both atom types of A and B belong to either Group I or Group II, it is a conjugated single (e.g. *cc-cc*, *cd-cd*)/aromatic single (e.g. *cp-cp*, *cq-cq*) bond; otherwise, it is a conjugated double (e.g. *cc-cd*, *c2-cc*, *c2-cd*)/aromatic double bond (e.g. *cp-cq*, *cp-ca*, *cq-ca*). As to *cg/ch*, only *cg-ch*, *cg-cl* and *ch-cl* are triple bonds and the other bonds that involve *cg* and *ch* are single bonds (e.g. *cl-cg*, *c2-cg*, *cc-cg*). More examples are shown in Fig. 5 to demonstrate the basic idea. Although this strategy works well for most of molecules, it fails in handling a certain kind of big ring systems with alternatively distributed single/double bonds. Nos. 20 and 21 in Fig. 5 are two examples of such unusual molecules. Fortunately, most of this kind of compounds are not drug-like and may not even be synthesizable at all.

In summary, an efficient algorithm is developed to assign force field atom types defined in a stand-alone atom type description table. With the first five features (Fields 1–5), one can define simple atom types; with the additional features of chemical environment and the bond type in a chemical environment (Fields 6–7), one can discriminate subtle local chemical environments and define complicated atom types.

#### 4. Applications in molecular topology generation

To describe an organic molecule or a repeated unit in a macromolecule in molecular mechanics, two types of information is needed: a set of force field parameters ( $K_r$ ,  $K_\theta$ ,  $V_n$ ,  $r_{eq}$ ,  $\theta_{eq}$ ,  $\gamma$ ,  $A_{ij}$ ,  $B_{ij}$ , etc., in Eq. (1)), and a residue topology that encodes the atomic connectivity, the atom type and the point charge of each atom. Fig. 6 shows the step-by-step procedures of generating system topologies that are read by *sander*, the program of running minimizations, molecular dynamics simulations in AMBER. First of all, either AM1-BCC or RESP charges are computed; then atom types are assigned; finally residue topology is generated with a program called *prepgen*. The step-by-step procedure can be simplified by running *antechamber*, the main program in the Antechamber module of the AMBER package. Like Babel [35], *antechamber* can perform molecular format transformations. It reads in and writes out several widely used molecular formats, including *prep* (AMBER residue topology file), *pdb*, *mol2*, *csd*, *mdl*, *hin* (HyperChem molecular format), *gcr*, *gzmat*, *gout* and *ac*. Unlike Babel, *antechamber* is not a pure molecular format conversion program: it also assigns atomic charges, atom types and bond types. The following charge methods are supported by *antechamber*: *RESP*, *AM1-BCC*, *CM1*, *CM2*, *Gasteiger* and *AM1 Mulliken*. By default, bond types are always assigned when *antechamber* reads in a molecule. The function of

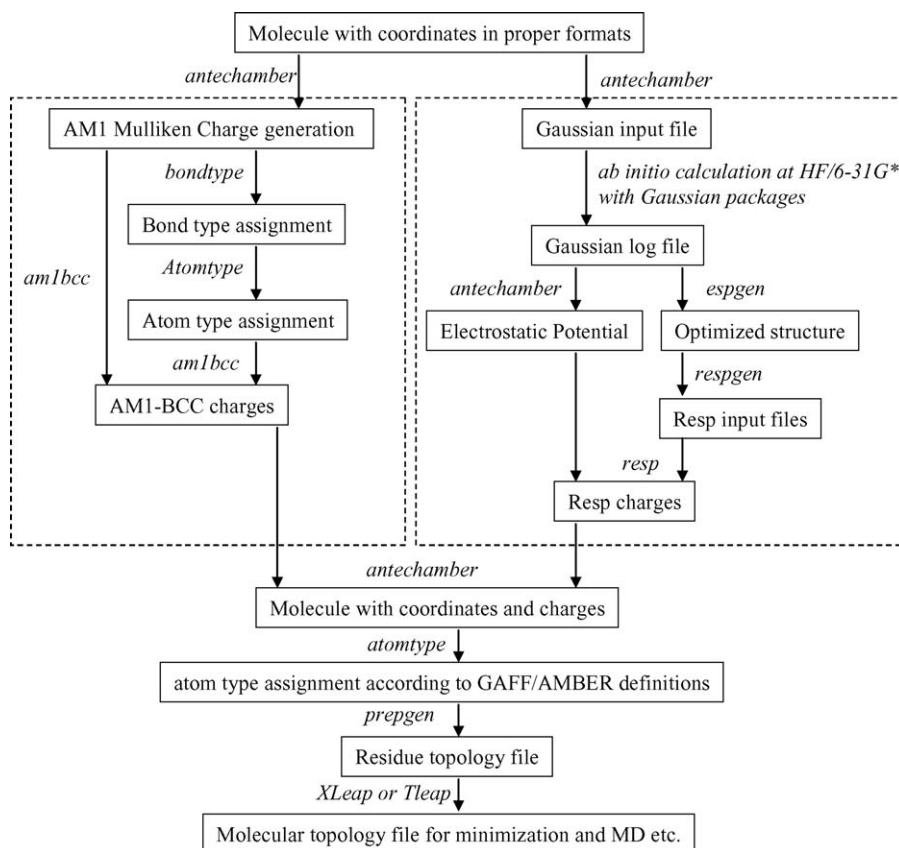


Fig. 6. A flowchart of applying the Antechamber package to generate AMBER system topologies. Basic steps of charge calculations are shown in dashed rectangles (AM1-BCC on the left and RESP on the right). The corresponding programs that fulfill certain functions are shown in italic.



*antechamber* can be fulfilled by calling individual Antechamber programs step by step as shown in Fig. 6.

It is common that some force field parameters are missing for some molecules even with GAFF. Therefore, another program called *parmchk* is applied to find the missing force field parameters and suggest substitutes based on the parameter similarity or calculate a new one based on empirical rules. The details on how *parmchk* works are beyond the scope of this paper.

The atom type and bond type perception algorithms have been extensively tested by 7722 molecules in a refined Comprehensive Medical Chemistry (CMC) database [36]. For each molecule, a *mol2* file was read in and a *prep* file was written out. The AM1-BCC bond types and GAFF atom types were assigned. The processing time was about 0.2 s per molecule. With a slightly improved version of Antechamber in AMBER8 [37], the bond types of 12 molecules could not be assigned correctly and the success rate was estimated to be 99.8%. However, with the current stand-alone version of Antechamber, both bond types and atom types of all the 7722 molecules could be successfully assigned without any error message.

## 5. Conclusions

In this work, we presented two algorithms of assigning atom types and bond types for fairly arbitrary organic molecules. The two algorithms are not only useful in molecular mechanics studies, but also in other applications such as QSPR model constructions when descriptors are atom type dependent. As an auxiliary module in the AMBER packages, Antechamber is mainly applied to generate topologies of arbitrary organic molecules for MM calculations. The combination of Antechamber with the general AMBER force field enables AMBER to study molecular structures and energies at a database level with a fashion of “on the fly”.

## Acknowledgements

This paper is dedicated to Peter A. Kollman, who initiated and directed this work prior to his death in May 2001. J.M.W. is grateful to acknowledge computer time at NCSA (MCB000013N (J. Wang, P. I.)). D.A.C. was supported by NIH grant GM-56531. The Antechamber program, which is licensed under GNU General Public Library, is available at <http://amber.scripps.edu/antechamber/antechamber.html>.

## References

- [1] W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, K.M. Merz Jr., D.M. Ferguson, D.C. Spellmeyer, T. Fox, J.W. Caldwell, P.A. Kollman, *J. Am. Chem. Soc.* 117 (1995) 5179–5197.
- [2] S.J. Weiner, P.A. Kollman, D.A. Case, U.C. Singh, C. Ghio, G. Alagona, S. Profeta Jr., P.J. Weiner, *Am. Chem. Soc.* 106 (1984) 765–784.

- [3] J. Wang, P. Cieplak, P.A. Kollman, *J. Comput. Chem.* 21 (2000) 1049–1074.
- [4] J. Wang, R.M. Wolf, J.W. Caldwell, P.A. Kollman, D.A. Case, *J. Comput. Chem.* 25 (2004) 1157–1174.
- [5] U. Burkert, N.L. Allinger, *Molecular Mechanics*, American Chemical Society, Washington, DC, 1982.
- [6] N.L. Allinger, *J. Am. Chem. Soc.* 99 (1977) 8127.
- [7] N.L. Allinger, Y.H. Yuh, J.-H. Lii, *J. Am. Chem. Soc.* 111 (1989) 8551–8566.
- [8] W.L. Jorgensen, J. Tirado-Rives, *J. Am. Chem. Soc.* 110 (1988) 1657–1666.
- [9] M. Clark, R.D. Cramer III, N. van Opdenbosch, *J. Comput. Chem.* 10 (1989) 982.
- [10] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, M. Karplus, *J. Comput. Chem.* 4 (1983) 187–217.
- [11] S. Lifson, A.T. Hagler, P. Dauber, *J. Am. Chem. Soc.* 101 (1979) 5111–5121.
- [12] M.J. Hwang, T.P. Stockfish, A.T. Halgler, *J. Am. Chem. Soc.* 116 (1994) 2515–2525.
- [13] F.M. Momany, R. Rone, *J. Comput. Chem.* 13 (1992) 888–900.
- [14] S.L. Mayo, B.D. Olafson, W.A. Goddard III, *J. Phys. Chem.* 94 (1990) 8879.
- [15] A.K. Rappé, C.J. Casewit, K.S. Colwell, W.A. Goddard III, W.M. Skiff, *J. Am. Chem. Soc.* 114 (1992) 10024–10035.
- [16] T.A. Halgren, *J. Comput. Chem.* 17 (1996) 490–519.
- [17] T.A. Halgren, *J. Comput. Chem.* 17 (1996) 520–552.
- [18] T.A. Halgren, *J. Comput. Chem.* 17 (1996) 553–586.
- [19] T.A. Halgren, *J. Comput. Chem.* 17 (1996) 587–615.
- [20] T.A. Halgren, *J. Comput. Chem.* 17 (1996) 615–641.
- [21] T.A. Halgren, *J. Comput. Chem.* 20 (1999) 730–748.
- [22] C.I. Bayly, P. Cieplak, W.D. Cornell, P.A. Kollman, *J. Phys. Chem.* 97 (1993) 10269–10280.
- [23] W.D. Cornell, P. Cieplak, C.I. Bayly, P.A. Kollman, *J. Am. Chem. Soc.* (1993) 9620–9631.
- [24] T. Fox, P.A. Kollman, *J. Phys. Chem.* 102 (1998) 8070–8079.
- [25] P. Cieplak, W.D. Cornell, C.I. Bayly, P.A. Kollman, *J. Comput. Chem.* 16 (1995) 1357–1377.
- [26] A. Jakalian, B.L. Bush, D.B. Jack, C.I. Bayly, *J. Comput. Chem.* 21 (2000) 132–146.
- [27] A. Jakalian, B.J. David, C.I. Bayly, *J. Comput. Chem.* 23 (2002) 1623–1641.
- [28] J. Wang, P.A. Kollman, *J. Comput. Chem.* 22 (2000) 1219–1227.
- [29] G. Liang, P.C. Fox, J.P. Bowen, *J. Comput. Chem.* 17 (1996) 940–953.
- [30] P.-O. Norrby, T. Liljefors, *J. Comput. Chem.* 19 (1998) 1146–1166.
- [31] R. Faller, H. Schmitz, O. Biermann, F. Muller-Plathe, *J. Comput. Chem.* 20 (1999) 1009–1017.
- [32] D. Weininger, *J. Chem. Inf. Comput. Sci.* 28 (1988) 31.
- [33] Sybyl user's manuals, Tripos Inc., St. Louis, 2000.
- [34] A. Pedretti, L. Villa, G. Vistoli, *Theor. Chem. Acc.* 109 (2003) 229–232.
- [35] P. Walters, M. Stahl, *Babel User's Manuals*, University of Arizona, Tucson, 1994.
- [36] MDL Information Systems Inc., 2004.
- [37] D.A. Case, T.A. Darden, T.E. Cheatham III, C.L. Simmerling, J. Wang, R.E. Duke, R. Luo, K.M. Merz, B. Wang, D.A. Pearlman, M. Crowley, S. Brozell, V. Tsui, H. Gohlke, J. Mongan, V. Hornak, G. Cui, P. Beroza, C. Schafmeister, J.W. Caldwell, W.S. Ross, P.A. Kollman, *AMBER 8*, University of California, San Francisco, 2004.

## Further reading

- [1] J. Wang, W. Wang, S. Huo, M. Lee, P.A. Kollman, *J. Phys. Chem.* 105 (2001) 5055–5067.