

Submitted to Journal of Chemical Information and Computer Sciences

Antechamber, An Accessory Software Package For Molecular Mechanical Calculations

Junmei Wang^{1*}, Wei Wang¹, Peter A. Kollman¹ and David A. Case²

¹. Department of Pharmaceutical Chemistry, University of California at San Francisco, San Francisco, California, 94143-0446

². Department of Molecular Biology, TPC15, The Scripps Research Institute, 10550 N. Torrey Pines Rd. La Jolla CA 92037

email: Junmei Wang (junmei@encysive.com)

David A. Case (case@scripps.edu)

*Correspondence to Junmei Wang and David A. Case

Funded by:

National Institutes of Health; Grant Number: GM-56531

Key Words:

Antechamber; Atom type recognition; Bond type recognition; Residue topology; Force field parameters; General AMBER force field (GAFF)

Abstract

We present a set of auxiliary programs for molecular mechanics studies on a wide class of organic molecules. This software package is devoted to solve the following problems: (1) identifying bond and atom types; (2) judging atomic equivalence; (3) generating residue topology files; and (4) finding missing force field parameters and supplying reasonable suggestions. As an accessory module in the AMBER 7 and AMBER 8 packages, *antechamber* can generate input automatically for most organic molecules in a database. The algorithms behind these manipulations are presented here, and may be useful outside the Amber family of programs as well.

1. Introduction

Molecular mechanics (MM) models are useful in studying structures, conformational energies, and other molecular properties, including vibrational frequencies, conformational entropies and dipole moments etc¹⁻²¹. Compared to first principles approaches, MM is less accurate than high-level *ab initio* models (such as G1, G2), but is much faster and simpler (since electrons are implicit in MM models) and has comparable accuracy to high-level quantum mechanical models in cases when the force field is well parameterized³. For large systems, such as proteins, DNA and RNA, MM model is almost the only choice, since calculations on these systems need huge computer resource and are very time-consuming. MM models also play an important role in the rational molecular design. Since both lead discovery and lead optimization need to screen a great amount of compounds, speed and automation are critical components of useful computational methods.

We have described elsewhere⁴ an organic force field (called the “general Amber force field”) which can describe most organic molecules in a fashion that is both reasonably accurate and is consistent with the way in which traditional Amber force fields describe proteins and nucleic acids. Here we describe the procedures and algorithms by which these parameters can be automatically assigned to arbitrary molecules, given only a three-dimensional structure as input. This set of procedures is called *antechamber* (roughly meaning “before Amber”), and are distributed as a new part of the Amber simulation package.

In the MM models, atoms that have similar properties (electronic, chemical and structural) share the same atom types. The parameters (bond stretching, angle bending, torsional twisting, van der Waals) are developed using the model molecules in order to reproduce the experimental and high-level *ab initio* data. Then the derived force field parameters are smoothly transferred to molecules outside the training set^{3,22-25}. Different force fields have different strategies to define the atom types. Therefore, in order to run MM calculations, one needs to recognize the atom types according to the specific definitions. Some MM software packages offer an in-core engine to perceive atom types and users cannot change the definitions at will. Other MM software packages do not have this kind engine and rely on user assigned atom types. It is thus useful to have algorithms to automatically recognize atom types that are both flexible and open to the user.

In this paper, we describe a simple algorithm for this purpose, where the atom types are described in a file that drives the atom type perception engine.

It is also useful to be able to describe atom equivalences in a flexible manner. For example, in AMBER force fields a restrained electrostatic potential (RESP)²⁶⁻²⁹ charge, which achieves a good performance in describing electrostatic field, is applied. During the RESP charge fitting, it is necessary to make the equivalent atoms bear the same charges since equivalent atoms are indiscriminable during the molecular dynamical simulations. For instance, the charges of three H in methyl and two H in methylene are equivalent. Our algorithm for identifying such equivalences, which are topological rather than structural, is also presented here.

Recently, Jakalian and Bayly developed a fast and efficient model AM1-BCC (AM1-bond charge correction) to generate the high-quality atomic charges resemble to the RESP charge³⁰⁻³¹. The AM1-BCC charges are actually the Mulliken charges at AM1 level with bond charge corrections (BCC). The BCC parameters are derived with a large training set in order to reproduce the RESP charges. AM1-BCC has a good performance in studying the molecular structure and conformational energy; therefore it is a good substitute of RESP in treating a great amount of compounds in the molecular design. In order to generate the AM1-BCC charges in a “on the fly” manner, one needs to recognize not only the atom types but also the bond types and the atomic equivalences automatically. In the antechamber package, we provided a program (am1bcc) to assign atom and bond types and then do BCC for any organic molecules, for which AM1-BCC has parameterized.

In the following, we give some details on the algorithms that include atom and bond type perception, atomic equivalence determination as well as the missing force field parameter assignment. The implementations of these algorithms in the antechamber package are also described. Most of the algorithms/programs described here require only basic molecular information (atomic numbers and bond connectivity) as input to work. Therefore, the algorithms may be useful in other MM code environments. The possible applications of the antechamber package are also discussed.

2. Basic algorithms

2.1 Atom equivalence perception

The bond connectivity information is the requisite of our atomic equivalence determination algorithm. The bond connectivity, if not being read in from the input, can be reliably derived based on atomic distances provided that the structures are quite reasonable (see 2.4.1). Among the widely used molecular formats, some include the bond connectivity information, such as mol2, csd; some do not, such as pdb, gaussian z-matrix.

First of all, we introduce the concept of atomic path, which is defined as a collection of a set of atoms (1, 2, 3, i-1, i, i+1 ...) in sequence. The neighboring atoms in a path (such as atoms i-1 and i, i and i+1) are bonded. The atomic path concept is extensively applied in this paper. The following is the procedure to judge the atomic equivalence based on the connectivity information. For each atom, all the possible paths that connect the current atom and all the terminal atoms are generated; then each path is evaluated using the following scoring function:

$$Score = \sum i \times 0.11 + an_i \times 0.08 \quad (I)$$

where i is the position index in the path and an_i is the atomic number of the atom at position i . The constants 0.11 and 0.08 in the above function are introduced in order to make the scores as diverse as possible, so that a path always has a unique score and a score always corresponds to a unique path. In the following, the scores are sorted from the smallest to the biggest. At last, the program compares any two sorted score arrays. If two arrays are identical, the two atoms are assumed equivalent. Figure 1 gives an example of how the algorithm finds the atomic equivalence for n-butane. Each atom in this molecule has 10 possible paths (since there are 10 terminal atoms in this molecule). Since the score array of C1 and C4 are identical, the two atoms are then equivalent. Similarly, C2 and C3, H4-H7, H1-H3 and H8-H10 are equivalent according to this algorithm.

2.2 Atom type perception

As we noted in the introduction section, atom types in MM are classified mainly based on electronic and structural properties, such as the atomic hybridization type, electron donor or acceptor capability, and so on. In antechamber, we apply the following features/fields to define

atom types: (1) atomic number; (2) number of connected atoms; (3) number of attached hydrogen atoms; (4) for hydrogen, number of the electron-withdrawal (*EW*) atoms connected to the atom that hydrogen attached; (5) atomic property, which refers to ring properties, aromatic properties and bond types (Seven bond types have been defined: single bond (*sb*), double bond (*db*), triple bond (*tb*), aromatic bond (AB), aromatic single bond (SB), aromatic double bond (DB) and delocalized bond (DL)); (6) subtle chemical environments; (7) bond connectivity in subtle chemical environment definitions. We believe that the above features/fields are sufficient to discriminate subtle chemical environment differences concisely and easily in most of cases. Table I lists the atom type descriptions for AMBER force fields¹⁻³; Table II, III and IV list the atom type descriptions for sulfur in GAFF⁴, carbons in MM3⁵⁻⁷ and nitrogen in MMFF94¹⁶⁻²¹, respectively. We want to emphasize here that the description files are totally independent of the core engine — the atom type-matching program. For the first four fields, the definitions are simple and straightforward. In the following paragraphs, we present some explanations on the last three fields and discuss how the atom type-assigning algorithm works in detail.

2.2.1 Ring system classification and aromatic type definition. In antechamber, 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 atom and tail atom are bonded. However, in this application only small rings (3-membered to 9-membered) are recognized. The following symbols are reserved to describe ring property: RG (any ring from 3-membered to 9-membered), RG3-RG9 (3-membered ring to 9-membered ring), NG (non-ring atom, actually including atoms in large rings (>9-membered)).

Antechamber defines five aromatic types for ring atoms, which are AR1 to AR5. AR1 is a pure aromatic atom, such as heavy atoms in benzene and pyridine; AR5 is a pure aliphatic ring atom; AR3 is a planar ring atom that has one or several double bonds formed with outside non-ring atoms; AR2 is a planar ring atom other than AR3; AR4 is a ring atom other than AR1, AR2, AR3 and AR5. Here a planar ring in antechamber is composed of C(X3), N(X2), N(X3), O(X2), S(X2), P(X2), P(X3), where X2 and X3 mean two and three bonded atoms. A planar ring usually belongs to one of two categories: forming an outside double bond (AR3) and having two continuous single bonds (AR2). Figure 2 shows some exemplary molecules that elucidate the definitions of ring types. In this figure, ring atoms in Molecules 3, 4a, 4b, 5, 6, 14a, 15b, 16a,

16b, 17a are AR1; 1, 2 are AR5; 7, 8, 9, 10, 17b, 18b, 19b are AR3; 11, 12, 13, 14b, 16c are AR2 and others are AR4.

Besides the ring and aromatic type properties, bond type information is also very important in molecular property definitions (2.2.2) and chemical environment definitions (2.2.3). The basic algorithm of bond type perception is presented in 2.3.

2.2.2 Atomic property definition. An atomic prosperity string, which combines the basic atomic properties (ring, aromatic and bond type properties) with logical “and” and logical “or”, is defined in a pair of middle-brackets. For example, [NR] stands for a non-ring system, [RG5.RG6,AR1.AR2.AR3] defines an 5 or 6-membered ring atom that has aromatic type of AR1 or AR2 or AR3. Here ‘.’ implies an “or” operation and ‘,’ an “and” operation.

2.2.3 Subtle chemical environment. In the chemical environment field, an atom usually takes a form of “*XXdd[ap]<an>*”, where “*XX*” is the name of an element, “*dd*” is a digital number, “*ap*” is the atomic property string of “*XX*” and “*an*” is the generic atom name of “*XX*”. It is notable that “*an*” is not the really atom name read from input file, but a generic symbol to discriminate atoms in chemical environment strings. “*sa*” is reserved as the generic symbol of the atom for which an atom type is being assigned. For example, C4 means a carbon with four connected atoms; C4[NR] means a non-ring sp^3 carbon; C3[AR1.AR2.AR3]) defines a sp^2 carbon in a planar ring system; N<n1> define a nitrogen with any number of connected atoms and having a generic name of “n1”. Each layer in the chemical environment chain needs a pair of brackets and atoms in the same layer are bracketed together. For example, (C(O1)) means the current atom is connected with a carbon and this carbon is connected to an oxygen that has only one connected atom. For this chemical environment string, “C” is in the first layer and “O1” is in the second layer. As another example, (C3(N3,N3)) means the current atom is connected with a sp^2 carbon, which is connected with two nitrogens that bind to three atoms including this carbon. For this string, “C3” is in the first layer and both the two “N3” are in the second layer.

In the following, we select some definitions from Tables I to IV and give some explanations. In the Row 6 of Table I the chemical environment string, (N2,N3(H)), defines two atomic paths: the current atom being connected to a nitrogen with two bonded atoms; the current atom being

connected to a nitrogen with three bonded atoms and one of them is hydrogen. In Row 8 of Table II, the chemical string (C3<a1>) and the bond connectivity field sa:a1:sb, define an atom path: the current atom being connected to an atom that has a generic atom name of “a1” with a single bond. In Row 12 of Table III, the chemical environment string (C3,O1) defines two atomic paths: the current atom being connected to a carbon with three bonded atoms; the current atom being connected to a terminal oxygen. In Row 23 of Table IV, the chemical environment string (C3(N3,N3)) defines two identical atomic paths: the current atom being connected to a carbon with three attached atoms and this carbon is connected to two nitrogens also with three attached atoms.

In fact, with both chemical environment and bond connectivity fields, one can discriminate very subtle chemical environment and define much more complicated and specific atom types than most of the entries listed in Tables I to IV. In the following, three more complicated examples are given for three molecules shown in Figure 3. For the first molecule, the methyl carbon in acetic acid can be defined as “(C3(O1[db],O2(H)))”, here “db” means “C3”, and “O1” forms a double bond. For the second molecule, a cycle-pentane carbon can be defined as “(C4<c1>(C4(C4(C4<c4>))))” with a bond connectivity field of “sa:c4:sb” or “(C4(C4<c2>),C4(C4<c3>))” with a bond connectivity field of “c2:c3:sb”. In the last example, the chemical environment of methyl carbon in the third molecule can be defined as “(C3(O1,C4<c1>(C4(C4<c3>(C4(C4<c6>))))),C4<c1>(C4(C4<c3>(C4 <c7>))))”. Atom “c6” and atom “c7” form a single bond (“c6:c7:sa”). The above definition has three separately chemical environment strings, which are C3-O1, C3-C4<c1>-C4-C4<c3>-C4-C4<c6> and C3-C4<c1>-C4-C4<c3>-C4<c7>. In the last two strings, the two atoms, c1 and c3, which splice the two paths, should be labeled explicitly.

Like the ring system detection, a recursive function is applied in chemical environment matching. All the possible atomic paths (not necessarily end up with terminal atoms) starting from the atom in question are first enumerated. Then the match between the chemical environment of the atom in question and that in the atom type definition table is checked. A successful match means not only all the atomic paths, but also related atom properties as well as bond connectivity defined in the last field are totally identical.

2.2.4 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 it is not '*' or '&', the reserved symbols in atom type definitions, which stand for the neglect of the current field and the end of the atom type definition, respectively.

For each atom, the match is performed for each definition string from the beginning and once a successful match is achieved, the atom type is assigned with the current atom type and the program jumps from the loop and is ready to handle another atom or halts. Therefore, the sequence order of the atom type definition is crucial for recognition. A thumb of rule is that the specific definitions should be placed before more general ones. If failure happens, the atom type perception program, *atomtype*, gives a warning message.

In order to make some definitions easier, wildcat atoms are introduced. For instance, one may define a wildcat atom XA1 that stands for either O1 or S1 and another wildcat atom XB stands for F, Cl, Br or I. Here, XA1 implies that it has only one bonded atom.

2.2.5 Atom type post-adjustment. It is often believed that bond types (single, double, triple, aromatic single, aromatic double, conjugated single and conjugated double) are essential in precisely describing subtle chemical environment. However, most of the force fields, including AMBER¹⁻³, only apply atom types for various reasons. In GAFF⁴, we also sacrificed bond type information for consistency to AMBER force fields and AMBER programs. In order to compensate the lack of atom types in GAFF, several pairs of special atom type pairs (ce/cf, cc/cd, ne/nf, nc/nd, pe/pf, pc/pd, cg/ch and cp/cq) were introduced to discriminate single and double bonds in aromatic and conjugated systems. The first atom type (ce, 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) is Group II atom type. The following is the basic rule: for an inner bond A-B in a conjugated/aromatic system, if atom types of A and B both belong to either Group I or Group II, it is a single bond; if not, it is a double bond or an aromatic bond (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 involving cg and ch are single bonds. Take a conjugated chain system as an example. The following bond are single bonds: cc-cc, cd-cd, cc-nc, cd-pd; on the other hand, cc-cg, cc-cd, c2-cc, cc-nd etc. are double

bonds. Figure 4 shows some examples on illuminating the basic algorithm. It is notable that this strategy, although works well for most of molecules, fails in handling molecules No. 20 and 21 in this figure. We found that most of problematic molecules have conjugated/aromatic rings attached to a big aliphatic ring ($10 + 4n$ ($n = 0, 1, 2 \dots$) membered rings). Fortunately, those compounds are not drug-like and may not exist and are not synthesizable at all.

Atom type perception is also an important issue in database searches. Smile and SLN (Sybyl Line Notation)³². are the two widely used atom type or chemical environment description tools in 2D-database searches. Unlike smile and SLN, our atom type description rules are specific for defining force field atom types, which can discriminate subtle chemical environment differences preciously and easily.

2.3 Bond type perception

In some situation, one needs to assign not only the atom types, but the bond types. This happens to the AM1-BCC charge model³⁰⁻³¹, whose BCC parameters are bond type dependent. Six bond types are defined in Reference 31, which are single, double, triple, aromatic single, aromatic double and delocalized bond. In this work, we introduced a simple algorithm to recognize the bond types in an automatic fashion.

2.3.1 Basic concepts. In the first place, we want to introduce three concepts. An atom's connectivity (*con*) is the number of atoms attached to it, for example the connectivity of a sp^3 carbon is 4 and that of a sp^2 carbon is 2. Connectivity information is provided by the input file in ac or mol2 format. Another concept, atomic valence (*av*) is defined by Formula II.

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

Where bo_i is bond order that takes a value of either 1 (single bond) or 2 (double bond) or 3 (triple bond). The third concept is atomic penalty score (*aps*) of *av*, which is 0 if the atomic valence of the concerned atom type is the most desirable. Table V lists the *aps* for seventeen atom types with different connectivity *con*. For example, for carbon, *aps* is 0 for if *av* equals to 4 and is 4 if

av equals to 3 or 5. The *aps* is either 0 or a positive integer. This feature is very important for the program to efficiently enumerate the valence states with the lowest penalty scores (*PS*), which is defined as the sum of the atomic penalty score-*aps*. We would like to point out that the atom types in Table V are totally different from the atom types in GAFF and AM1-BCC.

For most molecules, there are multiple valence states, since lots of atoms have more than one atomic valence. Each valence state has a penalty score. The smaller the *PS*, the more reasonable the structure is. Figure 5 shows some molecules in different valence states. In this figure, the abnormal atomic valences are labeled around the atoms that bear them. For each molecule, the valence state on the furthest left is the most reasonable assignment according to common sense, which also has the smallest penalty score. The total number of valence states can be enormous even for a small or middle-sized molecule. Suppose each atom has two atomic valences (*av*), the total number of valence states is 2^n , where n is the number of atoms. Therefore, it is important for us to filter out the lowest *PS* valence states efficiently.

2.3.2 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 that have a penalty score of n . This function is continually called for *PS* of n in ascending order starting from 0 until the number of accumulated valence states reaches a predefined limit, which is set to 100 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 to assign bond orders for a set of sorted valence states starting from the smallest penalty score until a successful assignment is achieved.

The following is details on how *boaf* works. For each bond, a bond order is assigned according to several basic rules. First of all, for each atom in a bond, if the bond order is determined, *con* is deducted 1 and *va* is deducted by the *bo*; If all the bonds are successfully assigned, *con* and *va* of each atom are both 0. In this situation 1 is returned and otherwise 0 is returned by *boaf*. Secondly, for one atom, if its *con* equals *av*, the bond order is set to 1; and if *con* equals to 1, the bond order is set to *av*. Finally, in the third rule, a try and error test is performed in case the bond

order is indeterminable: firstly, the program assumes the bond order is 1 and pursues bond order assignment according to the first two rules; if error happens, the bond order is reset to 2 and 3 finally. Here the error means *va* is not 0 when *con* is or *va* is 0 when *con* is not. If errors happen for all of the above three bond orders, the function exits and returns 0.

After the bond orders are determined, the 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 for others. According to the AM1-BCC's definitions, the aromatic atoms have AR1 or AR2 aromatic properties. Finally, C-O bonds in carboxyl anions and nitro groups are set to delocalized bond types.

2.3.3 The limitations. The bond orders in this algorithm are integers (that is, they are formal bond orders) and have different meanings from the bond orders 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 another word, there is more than one bond type assignment. In this situation, the bond type assignment is greatly affected by the sequence order of atoms in the input file. 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 noticeable that this algorithm may not work for molecules with unusual valence states, such as radicals using the default atomic penalty score (APS) listed in Table V. In this situation, the users may change the APS parameters to handle their specific molecules. Just as the atom type assignment program, *bondtype* is also a standalone program that can be integrated into other packages.

2.4 Parameter checking algorithm

A good force field should not only reliably transfer parameters from the training set to a test set, but also provide most, if not all force field parameters for the molecules it covers. In another word, a good force field should be transferable and complete. Unfortunately, it is a very big burden to develop a fully complete force field if it is not impossible. For example, in GAFF⁴,

which has 33 basic atom types, should have about 700 bond length types, 20,000 bond angle types and 1000, 000 torsional types if all the combinations are considered. Fortunately, for torsional angle, a general parameter strategy can efficiently reduce the parameter number to 200 or so. In GAFF, we developed 700 bond length parameters, 3000 bond angle parameters and 500 torsional angle parameters. Although bond lengths and torsional angles were full parameterized for the basic atom types, GAFF does need an empirical approach to estimate missing bond angle parameters. The program, Parmchk, serves this role in the antechamber package.

2.4.1 Atomic connectivity prediction. Atomic connectivity information is the prerequisite to enumerate the force field parameters (bond length, bond angle, torsional angle, improper dihedral angle and van der Waals) of a molecule. Some molecular formats include the connectivity information such as mol2 and csd; the others do not. For the later, the connectivity is automatically judged using the following algorithm. Suppose D_{ij} is the distance between any two atoms i, j and r_{ij} is the sum of the connectivity radii (α_i and α_j). The two atoms are assumed to be connected if D_{ij} is larger than $r_{ij}/2$ but smaller than r_{ij} plus an offset. Otherwise, they are disconnected. The offset is D_{ij} multiplying a constant, which is 0.15 if $D_{ij} \leq 1.5 \text{ \AA}$; 0.11 if $1.5 < D_{ij} \leq 1.9 \text{ \AA}$; 0.09 if $1.9 < D_{ij} \leq 2.05 \text{ \AA}$, and 0.08 if $D_{ij} > 2.05 \text{ \AA}$. The connectivity radii parameters, which are based on elements, are listed in Table VI. The connectivity radii parameters of H, C, N, O, S, P, F, Cl, Br, I, Fe, Zn, Cu, Mg and Mn are adopted from the MIDAS package³³ with minor adjustment. A good line correction is achieved between the atomic radii and the connectivity radii parameters for the 15 elements with the square of the correlation coefficient as 0.81. The connectivity radius parameters of other elements are estimated by atomic radii times 0.9396 (the slope) plus 0.006 (the interception) \AA .

2.4.2 Force field parameter checking. Once atomic connectivity is known, all the force field parameters of a molecule in question can be enumerated. For each force field parameter, parmchk first checks if it exists in the force field parameter file or not. If the parameter is missing, parmchk will find the most similar parameter available and use it instead. If the similarity search fails, the program turns to an empirical approach to calculate the parameters. Details on the empirical approach are available in Reference 4.

The atom type similarity is determined by the atom type corresponding table listed in Table VII. Each atom type has 1 to 3 corresponding atom types and one of them is the basic corresponding type, with which the generalized and replaced parameter has the highest chance to be parameterized. In the following, we take a bond angle A-B-C as an example to illuminate how the algorithm works. Suppose atom A has two corresponding atom types CA1 and CA2, and CA2 is the basic corresponding atom type; atoms B and C have one atom corresponding type each, which are CB and CC, respectively. If A-B-C is missing, the following substituted bond angle parameters are searched in sequence: CA1-B-C, CA2-B-C, A-CB-C, A-B-CC, CA1-CB-CC and CA2-CB-CC. The search stops once a substitute is found in the force field parameter set. If none of them is found, an empirical approach is applied to estimate the bond angle parameter.

3. A brief introduction of other important antechamber programs

So far, we have introduced three important programs in the antechamber package: `atomtype` for atom type perception, `bondtype` for bond type perception and `parmchk` for the missing force field parameter assignment. Besides these programs, `antechamber`, the main program of the package, and `prepgen`, the residue topology generation program, deserve a detailed introduction.

3.1 Antechamber – the main program

Similar to the widely used program `Babel`³⁴, `antechamber` can perform molecular format transformations. It can read in and write out several widely used molecular formats, including `prep` (amber residue topology file), `pdb`, `mol2`, `ac` (a new molecular format of antechamber), `csd`, `mdl`, `hin` (hyper molecular format), Gaussian cartesian and Z-matrix, Gaussian output (read in) etc. However, `antechamber`, unlike `Babel`, is not a pure molecular format conversion program. It can also assign atomic charges and atom types as well as bond types. The following charge methods are supported in `antechamber`: `RESP`, `AM1-BCC`, `CM2`, `Gasteiger` and `AM1 Mulliken`. These functions are fulfilled by calling other antechamber programs, such as `atomtype`, `bondtype` and `prepgen`.

In most of cases, `antechamber`, the main program of this package is recommended to generate the residue topology file because of its simplicity. However, on rare occasions, `antechamber` crashes and its core dumps. If this happens, one may correct the related file and run several individual programs to achieve his purpose. The program failures are caused mainly by the following

reasons: the atom connectivity is not correctly read in or predicted; the molecule has some unfilled valence; there are unrecognizable elements in the molecule, etc.

3.2 Prepgen

Prepgen is the program to generate residue topologies. It is straightforward to generate residue topologies using this program for the separated molecules, in another word, these molecules that have no covalent bonds to other residues. In the following, we will discuss the basic procedure and philosophy in generating amino acid-like residue topologies. 4-Hydroxyproline and 5-hydroxyproline shown in Figure 6 are taken as examples.

First of all, we need to design model molecules to mimic the real chemical environments of 4-hydroxyproline and 5-hydroxyproline in proteins. In Figure 6, the red part of each module molecule is the residue in question. CH₃CO in magenta and NHCH₃ in blue are added to mimic the chemical environment of hydroxyprolines in real proteins. In detail, the magenta C=O and CH₃ mimic the amide carbonyl group and α -carbon of the immediate previous residue, respectively; the blue NH and CH₃ mimic the amide nitrogen and α -carbon of the immediately followed residue, respectively.

Secondly, we need to derive the RESP charges for our model molecules. We perform *ab initio* minimizations on the model molecules at the HF/6-31G* level using Gaussian³⁵. In order to keep hydroxyprolines neutral, for each model molecule, we need to manually take the red part as a charge group and set the total charge to 0.0. Since we need manually revise the resp input files, antechamber itself alone cannot be used to generate the RESP charges. However, we can achieve the goal by running several individual programs. The basic procedure is shown in Figure 7.

Finally, we are ready to generate the residue topology file using prepgen. To generate an amino acid-like residue, which connects to and be connected to other residues, one needs an additional file to provide some additional information. In this file, we specify the red “N” being the head and the red “C” being the tail of the residue to be prepared. We also specify that CH₃CO in magenta and NHCH₃ in blue should be deleted. We also tell the program the atom types of magenta “C” and blue “N”, which directly connect to the residue in question. Provided the above

information, prepgen can generate the amino acid-like residue topology file in a straightforward fashion.

4. Applications of antechamber

4.1 Binding free energy calculations

Binding free energy calculations are an important topic in drug design. Molecular dynamics combined with MM-PB/SA has shown its power in predicting the binding free energy with a good balance between efficiency and accuracy³⁶⁻³⁹. Unlike another widely-used approach, linear interaction energy (LIE)⁴⁰⁻⁴⁴, MM-PBSA applies no empirical parameters at all for free energy calculations, which makes it a promising method to predict the binding free energy directly. To perform binding free energy calculations with MM-PBSA, one needs a set of files including residue topologies for the inhibitors, and additional force field files for the missing force field parameters, etc. Antechamber provides a set of programs to generate these files automatically. Figure 7 shows a flowchart to generate the residue and system topology files step by step. In fact, in most of cases the users may apply antechamber, the main program, to generate the residue topologies directly. However, this flowchart is helpful in case that antechamber crashes or if the users want to pursue some specific calculations. In both situations, one may resort to run a set of individual antechamber programs. Finally, it is noticeable that this procedure is not unchangeable and the users can use the antechamber programs in a flexible way to achieve their goals.

The following is the basic procedure of performing a MM-PBSA free energy calculation. First of all, the residue topology file and the additional force field parameter file are generated with antechamber and the system topology file is then generated with Leap in the AMBER package⁴⁵; then molecular minimizations followed by molecular dynamics simulations are carried out for structural relaxation and sampling; finally MM-PB/SA analysis is performed to calculate the binding free energy with the MM/PBSA module in the AMBER packages⁴⁵. If the solvation energies are calculated with Delphi⁴⁶⁻⁴⁷, one may apply delphigen, another program in the antechamber package to generate the charge and radius files.

4.2 Alanine Scanning

Alanine scanning and mutagenesis are important approaches to detect the key residues of an enzyme. If we assume the enzyme conformation does not change significantly upon the mutations, we may run only one MD simulation for the wild type, and then do residue mutations for each snapshot to generate proper coordinates. Then we can carry out MM-PB/SA analysis for both the wild type and the mutated species to find out which residues are “hot spots”. Take HIV-

1 RT/ EFAVIRENZ as an example. We can estimate the ΔG between the wild type and the observed mutation species (T267W) using the above analysis. If all the free energy differences are negative or small positive numbers, EFZ will have a good drug resistance property. Crdgrow, another program in the antechamber package can be applied to perform the residue mutations. This program reads in an incomplete pdb file (such as the main chain atoms of the residue to be mutated) as well as the residue topology file, and then generates a complete pdb file. Sometimes, a short minimization can make up the discrepancy caused by the mutations.

5. Conclusions

In this work, we presented a set of tools to generate input files for organic molecules, which can then be read into Leap or other molecular mechanics programs. Antechamber provides a set of algorithms to solve some important problems in MM calculations, such as atom type and bond type perceptions, and missing force field parameter substitutions and estimations. We provided several possible usages of the antechamber package in molecular mechanical studies, such as binding free energy calculation with MM-PBSA. Some tips on how to use antechamber efficiently were also introduced. Combined with the general AMBER force field, antechamber is a nice MM tool to study molecular structures and energies simply and automatically at the database level.

Acknowledgments

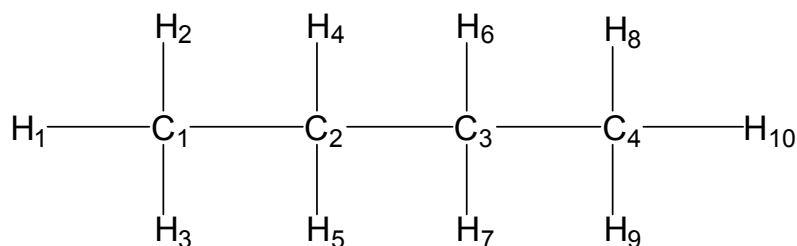
This paper is dedicated to Peter A. Kollman, who initialed and directed this work prior to his death in May 2001. J.M.W. is grateful to acknowledge computer time at NCSA (MCB000013N (J. Wang, P. I.)). Tutorials and the latest version of antechamber programs are provided from the antechamber webpage at <http://amber.scripps.edu/antechamber>.

References

1. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz Jr., K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. *J. Am. Chem. Soc.* **1995**, 117, 5179-5197.
2. Weiner, S. J.; Kollman, P. A.; Case, D. A.; Singh, U. C.; Ghio, C.; Alagona, G.; Profeta, Jr., S.; Weiner, P. *J. Am. Chem. Soc.* **1984**, 106, 765-784.
3. Wang, J.; Cieplak, P.; Kollman, P. A. *J. Comput. Chem.* **2000**, 21, 1049-1074.
4. Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. *J. Comput. Chem.* in press.
5. Burkert, U.; Allinger, N. L. *Molecular Mechanics*, American Chemical Society: Washington, D. C., **1982**.
6. Allinger, N. L. *J. Am. Chem. Soc.* **1977**, 99, 8127
7. Allinger, N. L.; Yuh, Y. H.; Lii, J.-H. *J. Am. Chem. Soc.* **1989**, 111, 8551-8566
8. Jorgensen, W. L.; Tirado-Rives, J. *J. Am. Chem. Soc.* **1988**, 110, 1657-1666
9. Clark, M.; Cramer, R. D. III; van Opdenbosch, N. *J. Comput. Chem.* **1989**, 10, 982.
10. Brooks, B. R.; Brucoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. *J. Comput. Chem.* **1983**, 4, 187-217.
11. Lifson, S.; Hagler, A. T.; Dauber, P. *J. Am. Chem. Soc.* **1979**, 101, 5111-5121
12. Hwang, M. J.; Stockfish, T. P.; Halgler, A. T. *J. Am. Chem. Soc.* **1994**, 116, 2515-2525
13. Momany, F. M.; Rone, R. *J. Comput. Chem.* **1992**, 13, 888-900.
14. Mayo, S. L.; Olafson, B. D.; Goddard III, W. A. *J. Phys. Chem.* **1990**, 94, 8879.
15. Rappé, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard III, W. A.; Skiff, W. M. *J. Am. Chem. Soc.* **1992**, 114, 10024-10035.
16. Halgren, T. A. *J. Comput. Chem.* **1996**, 17, 490-519.
17. Halgren, T. A. *J. Comput. Chem.* **1996**, 17, 520-552.
18. Halgren, T. A. *J. Comput. Chem.* **1996**, 17, 553-586.
19. Halgren, T. A. *J. Comput. Chem.* **1996**, 17, 587-615.
20. Halgren, T. A. *J. Comput. Chem.* **1996**, 17, 615-641.
21. Halgren, T. A. *J. Comput. Chem.* **1999**, 20, 730-748.
22. Wang, J.; Kollman, P. A. *J. Comput. Chem.* **2000**, **22**, 1219-1227.
23. Liang, G.; Fox, P. C.; Bowen, J. P. *J. Comput. Chem.*, **1996**, 17, 940-953.
24. Norrby, P.-O.; Liljefors, T. *J. Comput. Chem.*, **1998**, 19, 1146-1166.

25. Faller, R.; Schmitz, H.; Biermann, O.; Muller-Plathe, F. *J. Comput. Chem.*, **1999**, 20, 1009-1017.
26. Bayly, C. I.; Cieplak, P.; Cornell, W. D.; Kollman, P. A. *J. Phys. Chem.* **1993**, 97, 10269-10280.
27. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Kollman, P. A. *J. Am. Chem. Soc.* **1993**, 9620-9631.
28. Fox, T.; Kollman, P. A. *J. Phys. Chem.* **1998**, 102, 8070-8079.
29. Cieplak, P.; Cornell, W. D.; Bayly, C. I.; Kollman, P. A. *J. Comput. Chem.* **1995**, 16, 1357-1377.
30. Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I. *J. Comput. Chem.*, **2000**, 21, 132-146.
31. Jakalian, A.; David, B. J.; Bayly, C. I. *J. Comput. Chem.*, **2002**, 23, 1623-1641.
32. Sybyl user's manuals, Tripos Inc., St. Louis, **2000**.
33. Midas user's manual, University of California, San Francisco, **1997**.
34. Walters, P.; Stahl, M. Babel user's manuals, University of Arizona, Tucson, **1994**.
35. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Zakrzewski, V. G.; Montgomery, Jr., J. A.; Stratmann, R. E.; Burant, J. C.; Dapprich, S.; Millam, J. M.; Daniels, A. D.; Kudin, K. N.; Strain, M. C.; Farkas, O.; Tomasi, J.; Barone, V.; Cossi, M.; Cammi, R.; Mennucci, B.; Pomelli, C.; Adamo, C.; Clifford, S.; Ochterski, J.; Petersson, G. A.; Ayala, P. Y.; Cui, Q.; Morokuma, K.; Malick, D. K.; Rabuck, A. D.; Raghavachari, K.; Foresman, J. B.; Cioslowski, J.; Ortiz, J. V.; Baboul, A. G.; Stefanov, B. B.; Liu, G.; Liashenko, A.; Piskorz, P.; Komaromi, I.; Gomperts, R.; Martin, R. L.; Fox, D. J.; Keith, T.; Al-Laham, M. A.; Peng, C. Y.; Nanayakkara, A.; Gonzalez, C.; Challacombe, M.; Gill, P. M. W.; Johnson, B. G.; Chen, W.; Wong, M. W.; Andres, J. L.; Head-Gordon, M.; Replogle, E. S. and Pople, J. A. Gaussian, Inc., Pittsburgh PA, 1998.
36. Srinivasan, J.; Miller, J.; Kollman, P. A.; Case, D. A. *J. Biol. Struct. Dyn.*, **1998**, 16, 671-682.
37. Wang, J.; Morin, P.; Wang, W.; Kollman, P. A. *J. Am. Chem. Soc.*, **2001**, 123, 5221-5230.
38. Huo, S.; Wang, J.; Cieplak, P.; Kollman, P. A.; Kuntz, I. D. *Journal of Medicinal Chemistry*, **2002**, 45, 1412-1419.
39. Wang, W.; Lim, W.; Jakalian, A.; Wang, J.; Luo, R.; Bayly, C.; Kollman, P. A. *J. Am. Chem. Soc.*, **2001**, 123, 3986-3994.
40. Åqvist, J.; Medina, C.; Samuelsson, J. *Protein Engineering*, **1994**, 7, 385-391.
41. Hansson, T.; Åqvist, J. *Protein Engineering*, **1995**, 8, 1137-1144.
42. Åqvist, J. *J. Comput. Chem.* **1996**, 17, 1587-1597.

43. Wang, W.; Wang, J.; Kollman, P. A. *Proteins: Struct. Funct. & Gene.* **1999**, 34, 395-402.
44. Smith, R. H.; Jorgensen, W. L.; Tirado-Rives, J.; Lamb, M. L.; Janssen, P. A. J.; Michejda, C. J.; Smith, B. K. *J. Med. Chem.* **1998**, 41, 5272-5286.
45. Case, D. A.; Pearlman, D. A.; Caldwell, J. W.; Cheatham III, T. E.; Wang, J.; Ross, W. S.; Simmerling, C. L.; Darden, T. A.; Merz, K. M.; Stanton, R. V.; Cheng, A. L.; Vincent, J. J.; Vincent, M.; Crowley, V.; Tsui, V.; Gohlke, H.; Radmer, R. J.; Duan, Y.; Pitera, J.; Massova, I.; Seibel, G. L.; Singh, U. C.; Weiner, P. K. and Kollman, P. A. *AMBER7*, University of California, San Francisco, 2002.
46. Honig, B.; Nicholls, A. *Science*, **1995**, 268, 1144-1149.
47. Gilson, M. K.; Honig, B. *Proteins*, **1998**, 4, 7-18.



Paths of atom C1:

- 1 C1-H1 $(1 \times 0.11 + 0.08 \times 6) + (2 \times 0.11 + 0.08 \times 1) = 0.89$
- 2 C1-H2 $(1 \times 0.11 + 0.08 \times 6) + (2 \times 0.11 + 0.08 \times 1) = 0.89$
- 3 C1-H3 $(1 \times 0.11 + 0.08 \times 6) + (2 \times 0.11 + 0.08 \times 1) = 0.89$
- 4 C1-C2-H4 ...
- 5 C1-C2-H5 ...
- 6 C1-C2-C3-H6 ...
- 7 C1-C2-C3-H7 ...
- 8 C1-C2-C3-C4-H8 ...
- 9 C1-C2-C3-C4-H8 ...
- 10 C1-C2-C3-C4-H8 $(1 \times 0.11 + 0.86 \times 6) + (2 \times 0.11 + 0.08 \times 6) + (3 \times 0.11 + 0.08 \times 6) + (4 \times 0.11 + 0.08 \times 6) + (5 \times 0.11 + 0.08 \times 1) = 3.65$

Sorted score table

	1	2	3	4	5	6	7	8	9	10
1 C1	0.89	0.89	0.89	1.70	1.70	2.62	2.62	3.65	3.65	3.65
2 C2	0.89	0.89	1.70	1.70	1.70	1.70	1.70	2.62	2.62	2.62
3 C3	0.89	0.89	1.70	1.70	1.70	1.70	1.70	2.62	2.62	2.62
4 C4	0.89	0.89	0.89	1.70	1.70	2.62	2.62	3.65	3.65	3.65
5 H1	0.19	1.30	1.30	2.22	2.22	3.25	3.25	4.39	4.39	4.39
6 H2	0.19	1.30	1.30	2.22	2.22	3.25	3.25	4.39	4.39	4.39
7 H3	0.19	1.30	1.30	2.22	2.22	3.25	3.25	4.39	4.39	4.39
8 H4	0.19	1.30	2.22	2.22	2.22	2.22	2.22	3.25	3.25	3.25
9 H5	0.19	1.30	2.22	2.22	2.22	2.22	2.22	3.25	3.25	3.25

10	H6	0.19	1.30	2.22	2.22	2.22	2.22	2.22	3.25	3.25	3.25
11	H7	0.19	1.30	2.22	2.22	2.22	2.22	2.22	3.25	3.25	3.25
12	H8	0.19	1.30	1.30	2.22	2.22	3.25	3.25	4.39	4.39	4.39
13	H9	0.19	1.30	1.30	2.22	2.22	3.25	3.25	4.39	4.39	4.39
14	H10	0.19	1.30	1.30	2.22	2.22	3.25	3.25	4.39	4.39	4.39

Figure 1. Atomic equivalence perception for n-butane. The ten paths of atom C1 are shown in the middle part of this figure and the sorted array for each atom is shown at the bottom. From this table, one can see that score arrays of C1 and C4, C2 and C3, H4-H7, H1-H3, and H8-H10 are identical.

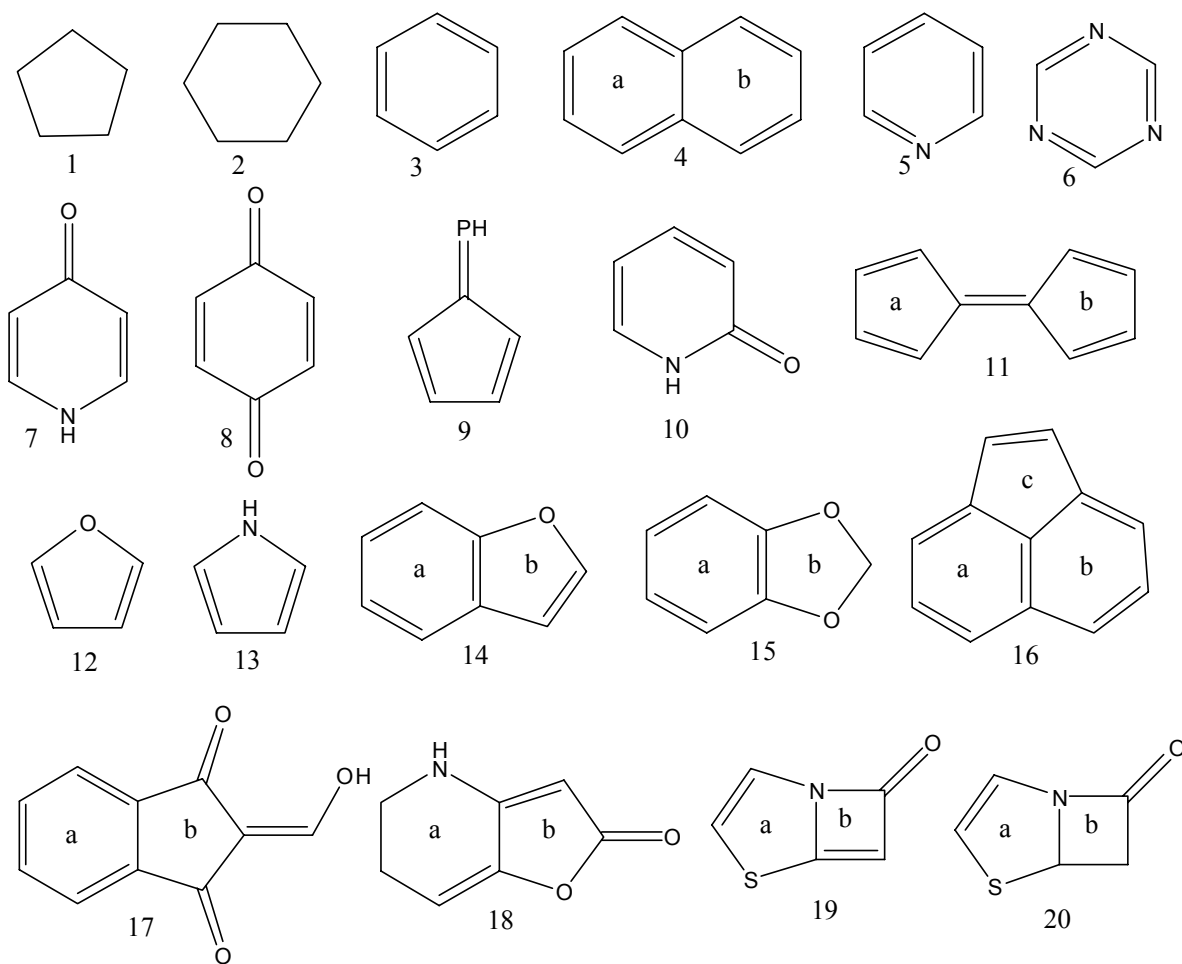


Figure 2. Exemplary molecules that elucidate the definitions of aromatic types in Antechamber. Ring atoms in Molecules 3, 4a, 4b, 5, 6, 14a, 15b, 16a, 16b, 17a are AR1; 1, 2 are AR5; 7, 8, 9, 10, 17b, 18b, 19b are AR3; 11, 12, 13, 14b, 16c are AR2, and others are AR4.

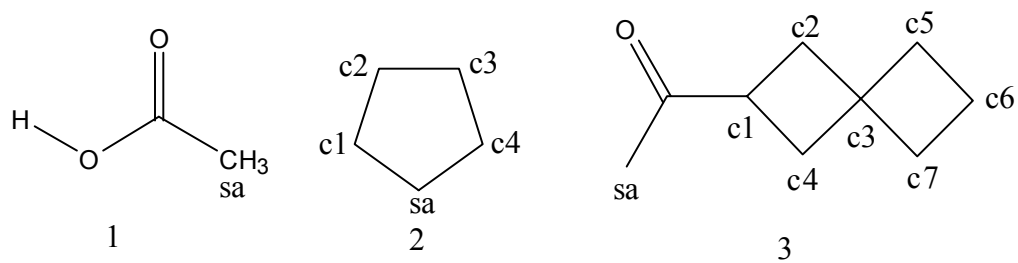


Figure 3. Exemplary molecules that elucidate the definitions of chemical environment strings. “sa” is the atom for which an atom type is assigned and c1-c7 are generic atom names.

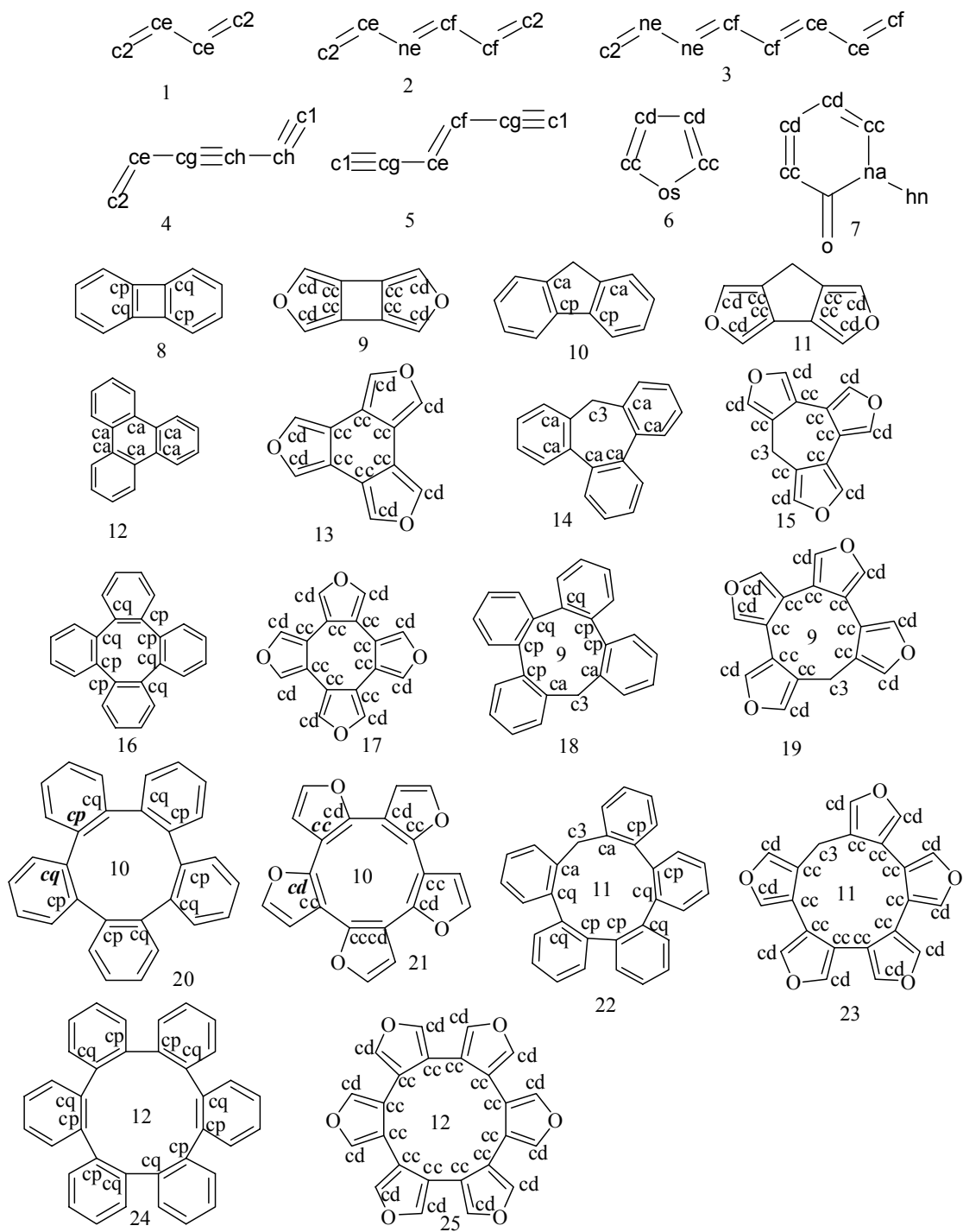
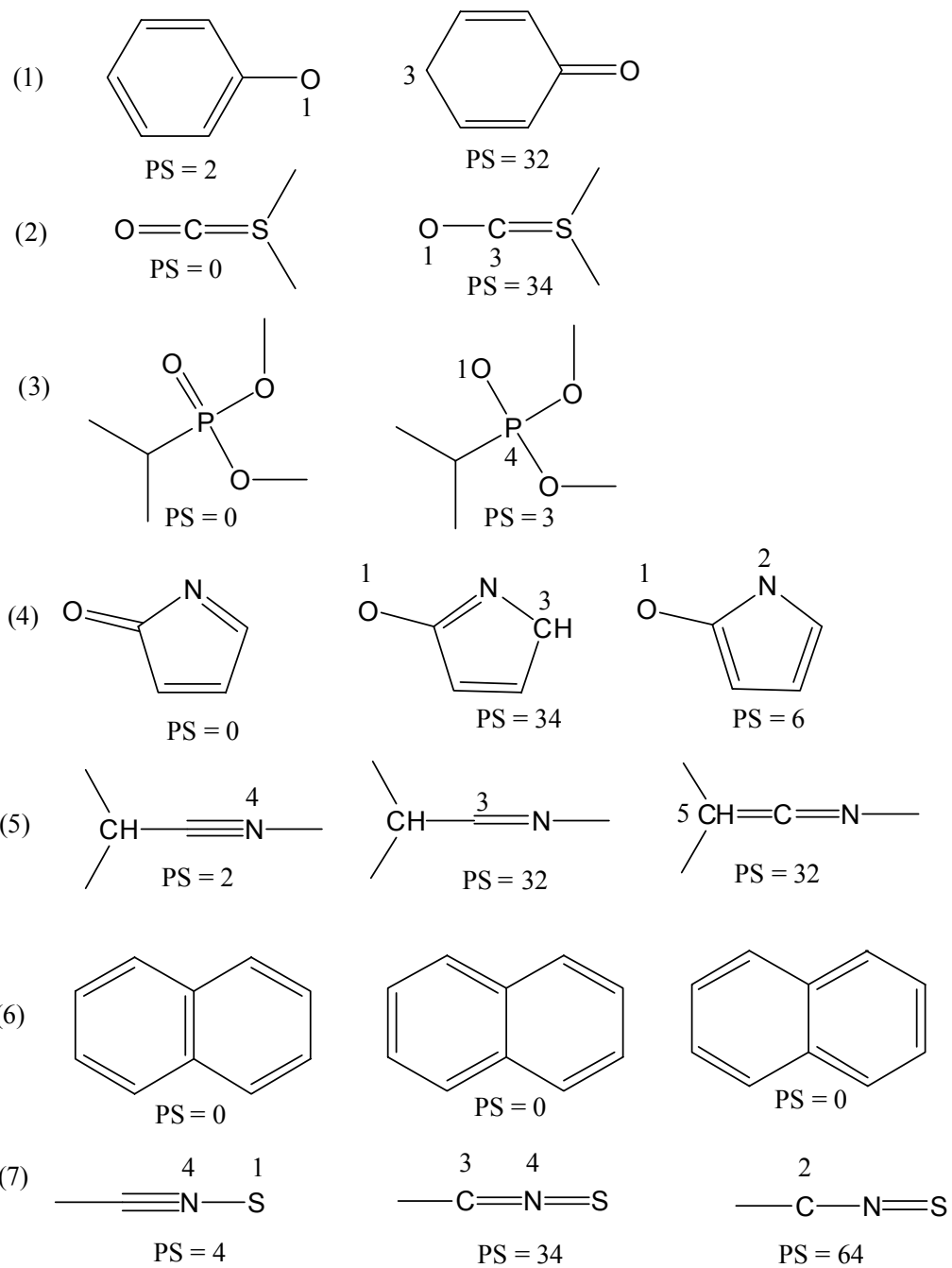


Figure 4. A simple algorithm to discriminate single and double bonds in conjugated and bi-phenyl systems. For No. 20 and 21, our strategy fails to assign atom types correctly and atom types that cause the failure are marked with bold italic font.



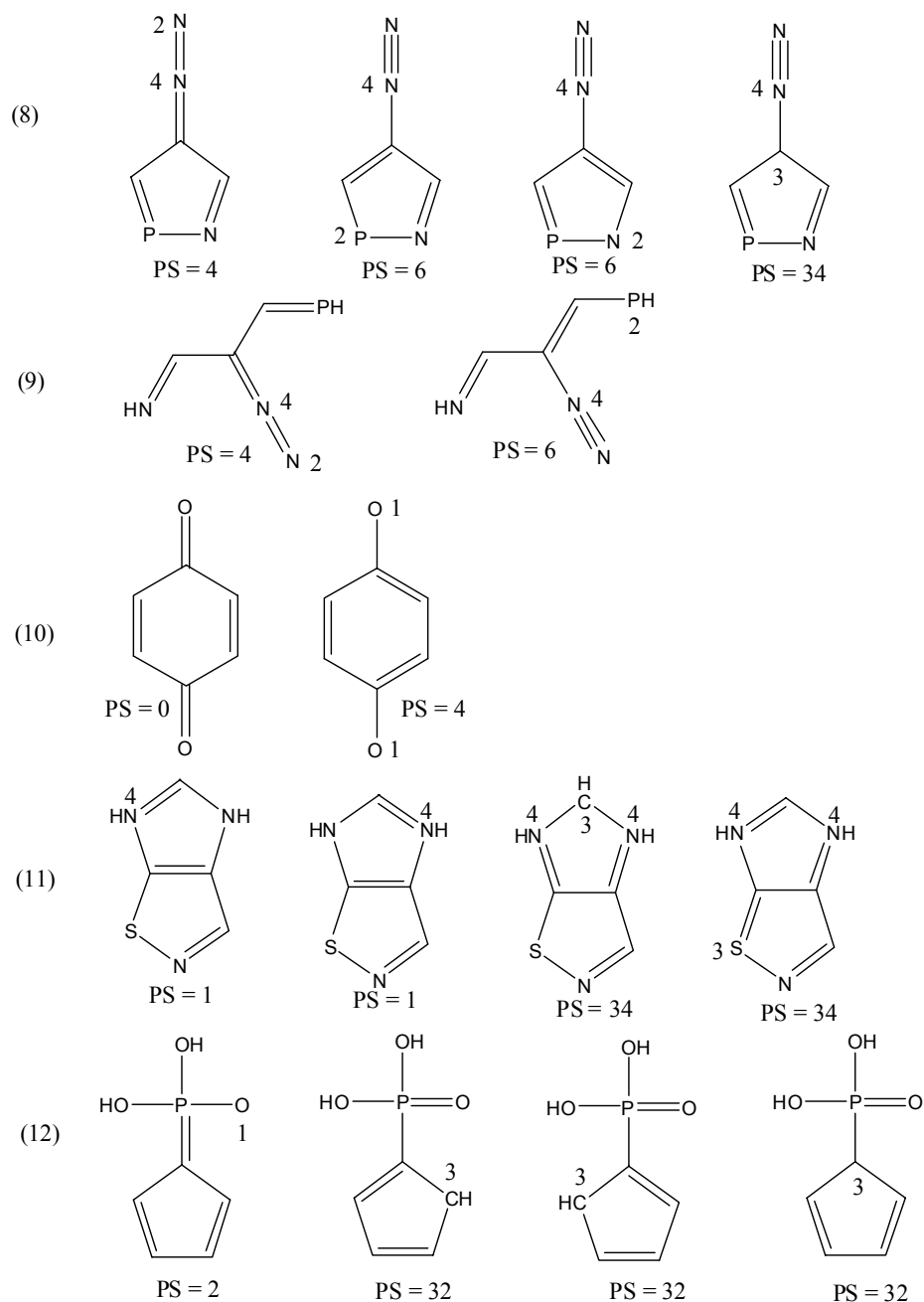
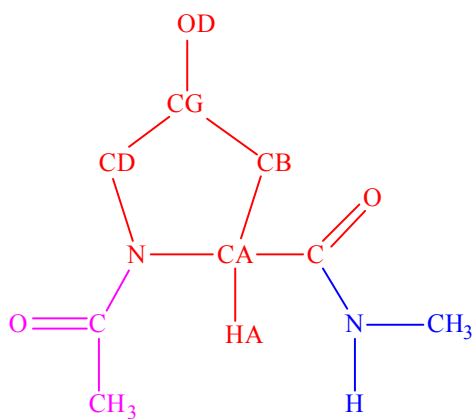
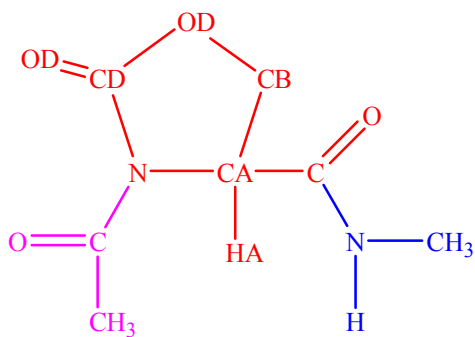


Figure 5. Exemplary molecules that elucidate the mechanism of bond type prediction. Abnormal atomic valences are given indications near those atoms. For each molecule, several possible bond order assignments are shown with their penalty scores (PS). The most left one is the most reasonable assignment according to the common sense. The first 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.



(a)



(b)

Figure 6. Preparation of residue topology files for 4-hydroxyproline (a) and 5-hydroxyproline (b). The red part of this molecule is the real part in the residue topology, other parts (CH_3CO in magenta and NHCH_3 in blue) are used to mimic the chemical environment of ALA in real proteins.

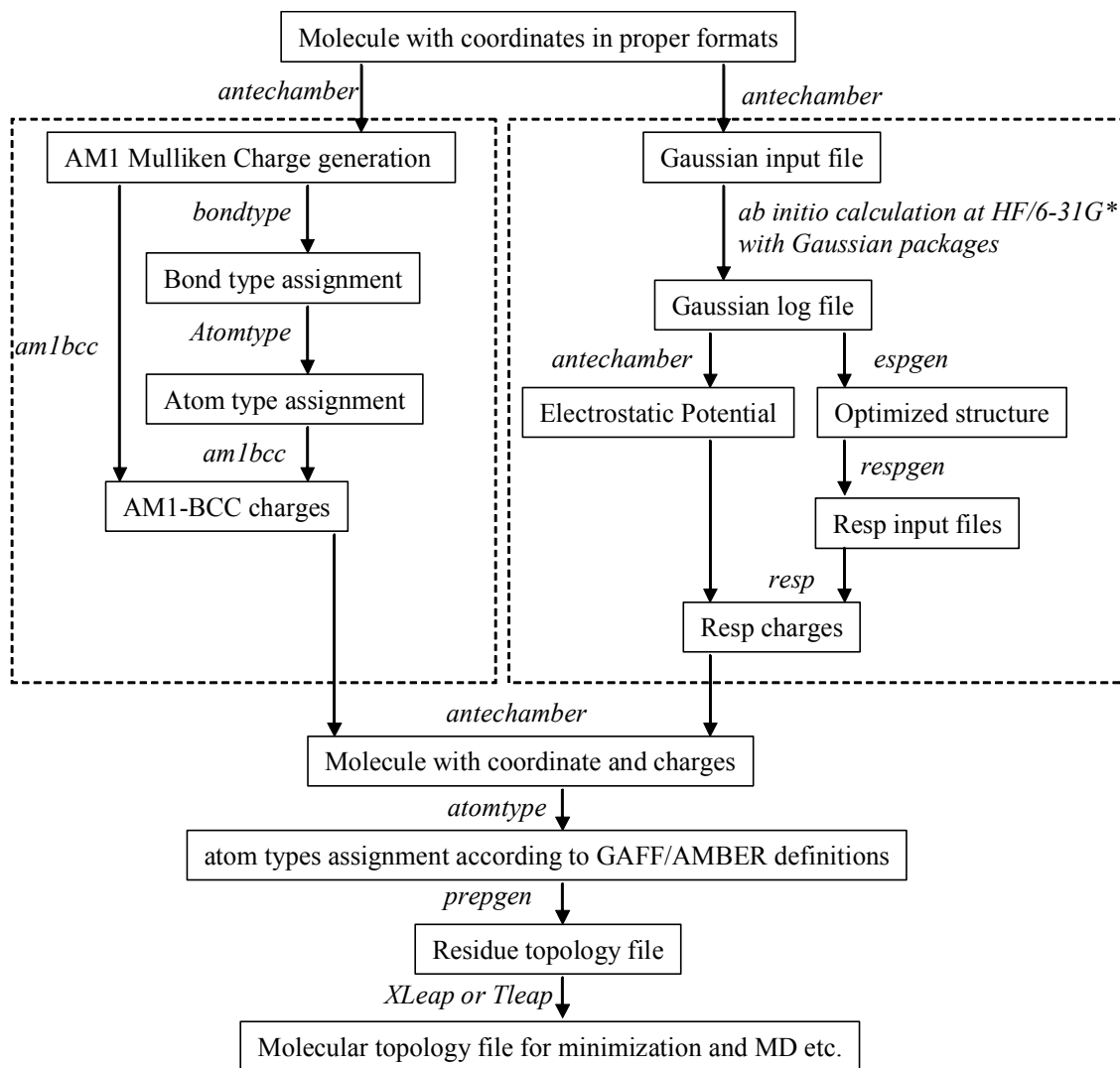


Figure 7. The flowchart of applying the antechamber package to generate the AMBER residue topology file step by step. Basic procedures of charge calculations are shown in the dashed rectangles (AM1-BCC on the left and RESP on the right).

Table I An Atom Type Description File For The AMBER Force Field. F1-F6 Are Electronic/Structural Features Described As The Following. 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 Immediate Connected Atom; F5: Atomic Propriety; F6: Chemical Environment Definition; F7: Bond Connectivity In Chemical Environment Defined In F6. Some Wildcat 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; In Antechamber, There Are Eight Bond Types 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), db(double bond, including aromatic double).

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	&						

Table II Descriptions Of The Sulfur Atom Types For The General AMBER Force Field (GAFF).
F1-F7 And Wildcat Elements Are Defined in Table I.

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	&				

Table III Descriptions Of The Carbon Atom Types In MM3*. F1-F7 And Wildcat Elements Are Defined In Table I..

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	&				

* atom type 29 (carbon radical), 30 (carbonium ion), 113 (c in ferrocene-H) and 114 (carbon in ferrocene-C) do not include.

Table IV Descriptions Of The Nitrogen Atom Types In MMFF. F1-F7 And Wildcat Elements Are Defined In Table I.

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	&				

Table V Atomic Penalty Scores (*aps*) For Different Atom Types In Bond Type Perception. In This Table, AV0-AV6 Stand For Atomic Valence 0 to 6. In The Atom Type Column, X1-X6 Indicate That The Number Of Connected Atoms Is 1 to 6.

No	Atom type	AV0	AV1	AV2	AV3	AV4	AV5	AV6
1	H, F, Cl, Br, I	64	0	64				
2	C			64	32	0	32	64
7	C in COO ⁻				64	0	32	64
3	Si					0		
3	N (X1)			2	0	32		
4	N (X2)			4	0	2		
5	N (X3)			32	0	1	2	
6	N (X4)			64	0	64		
7	N in Nitro				64	0	32	64
8	O (X1)		2	0	64			
9	O (X2)		32	0	64			
10	P (X1)			2	0	32		
11	P (X2)			4	0	2		
12	P (X3)			32	0	1	2	
13	P (X4)				64	1	0	32
14	S (X1)		2	0	64			
15	S (X2)		32	0	32	1		
16	S (X3)				1	0	2	2
17	S (X4)					4	2	0

Table VI List Of The Atomic Connectivity Parameters (Å). The Atomic Radii (Å) Are Adopted From <http://www.chemicool.com/>. The Elements That Have Connectivity Radius Parameters From Midas³² Are Italicized.

No	Element	Valence	Atomic radius	Parameter	No	Element	Valence	Atomic radius	Parameter
1	<i>H</i>	0	0.371	0.35	2	Li	+1	0.9	0.85
3	Be	+2	0.59	0.56	4	B	+3	0.41	0.39
5	<i>C</i>	0	0.772	0.73	6	<i>N</i>	0	0.7	0.66
7	<i>O</i>	0	0.73	0.69	8	<i>F</i>	0	0.72	0.68
9	Na	+1	1.16	1.10	10	<i>Mg</i>	+2	0.86	0.81
11	Al	+3	0.675	0.64	12	Si	0	1.176	1.11
13	<i>P</i>	0	1.1	1.04	14	<i>S</i>	0	1.02	0.96
15	<i>Cl</i>	0	1	0.95	16	K	+1	1.52	1.43
17	Ca	+2	1.14	1.08	18	Sc	+3	0.885	0.84
19	Ti	+3	0.81	0.77	20	V	+3	0.78	0.74
21	Cr	+3	0.755	0.72	22	<i>Mn</i>	+2	0.89	0.84
23	<i>Fe</i>	+3	0.738	0.70	24	Co	+2	0.838	0.79
25	Ni	+2	0.83	0.79	26	<i>Cu</i>	+2	0.87	0.82
27	<i>Zn</i>	+2	0.88	0.83	28	Ga	+3	0.76	0.72
29	Ge	+2	0.87	0.82	30	As	0	1.2	1.13
31	Se	0	1.19	1.12	32	<i>Br</i>	0	1.14	1.08
33	Sr	+2	1.32	1.25	34	Ru	+3	0.82	0.78
35	Rh	+3	0.805	0.76	36	Pd	+2	1	0.95
37	Ag	+1	1.08	1.02	38	Cd	+2	1.09	1.03
39	<i>I</i>	0	1.33	1.26	40	Ba	+2	1.49	1.41
41	Pt	+2	0.94	0.89	42	Au	+1	1.51	1.42
43	Hg	+2	1.33	1.26	44	Tl	+1	1.64	1.55
45	Pb	+2	1.33	1.26					

Table VII Atom Corresponding Types In GAFF. The Following Basic Atom Types Have Their Corresponding Types Identical To Themselves: hc, ho, hp, hw, hs, hn, c1, c2, c3, n1, n2, n3, n4, o, oh, os, s, ss, sh, s4, s6, p2, p3, p4, p5. If There Are More Than One Corresponding Atom Type, the One In Parenthesis Is the Basic Corresponding Atom Type.

No	Types	Corresponding Type	No	Types	Corresponding Type
1	h1-h3, ha	<u>hc</u>	2	h4, h5	ha (hc)
3	c	C2	4	ca	<u>c2</u>
5	cc, cd	ce (c2)	6	ce, cf	c2
7	cg, ch	<u>C1</u>	8	cp	ca (c2)
9	cu, cv	C2	10	cx, cy	<u>c3</u>
11	na	N3	12	nb	nc (n2)
13	nc, nd	<u>ne (n2)</u>	14	ne, nf	n2
15	nh, no	N3	16	sx	<u>s4</u>
17	sy	S6	18	pb	pc (p2)
19	pc	<u>Pe (p2)</u>	20	pd	pe (p2)
21	pe	p2	22	pf	<u>p2</u>
23	px	p3	24	py	p4