PNNL - 13850

HOSTDESIGNER

USER'S MANUAL

Version 1.0

April 2002

DR. BENJAMIN P. HAY AND DR. TIMOTHY K. FIRMAN

THEORY, MODELING, AND SIMULATION GROUP

W. R. WILEY ENVIRONMENTAL MOLECULAR SCIENCES LABORATORY

PACIFIC NORTHWEST NATIONAL LABORATORY

PO BOX 999, RICHLAND, WASHINGTON 99352

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ACKNOWLEDGEMENTS

This research was funded in part by the Chemical Sciences, Office of Basic Energy Sciences, Office of Science, U. S. Department of Energy (DOE) and in part by the Laboratory Directed Research and Development Program, Fundamental Science Division of the Pacific Northwest National Laboratory (PNNL). The research was performed in the William R. Wiley Environmental Molecular Sciences Laboratory (EMSL) at PNNL. The EMSL is a national user facility funded by the Office of Biological and Environmental Research in the DOE. The authors express their appreciation to Dr. J. B. Nicholas (Genentech, Inc.) for his considerable input in the early design of this software, Prof. J. W. Ponder (Washington University Medical School) for granting permission to distribute portions of the TINKER code used in the development of the OVERLAY algorithm, and Drs. T.P. Straatsma and W. deJong (PNNL) for their assistance with writing and debugging the Fortran code.

TABLE OF CONTENTS

DISCLAIMER 2

ACKNOWLEDGEMENTS 2

TABLE OF CONTENTS 3

1.0 INTRODUCTION 4

2.0 DESCRIPTION OF THE ALGORITHMS 6

2.1 THE LINKER ALGORITHM 6

2.2 THE OVERLAY ALGORITHM 13

2.3 THE LINK LIBRARY 14

3.0 HOW TO DRIVE HOSTDESIGNER 21

3.1 INSTALLATION 21

3.1 THE FILE NAMED “CONTROL” 21

3.2 OTHER INPUT FILES FOR THE LINKER MODE 24

3.3 OTHER INPUT FILES FOR THE OVERLAY MODE 26

3.4 DESCRIPTION OF OUTPUT FILES 28

4.0 HOW TO CITE HOSTDESIGNER IN THE LITERATURE 30

1.0 INTRODUCTION

Because the ability to successfully design effective and selective metal ion host molecules would have a significant impact in many scientific fields, there is a large body of research focused on understanding the nature of host-guest interactions. Fundamental studies of the interactions between a metal ion and simple coordinating groups, such as ether oxygens, amine nitrogens, or arenes, have elucidated geometric features that lead to optimal binding. It has been shown that this type of information can be incorporated into force field models to provide a rapid method of screening proposed host structures, in other words, rank-ordering a set of hosts in terms of their binding affinity for a given guest. It also has been demonstrated that more accurate, but more costly, screening can be accomplished with molecular dynamics or electronic structure calculations. However, although understanding the nature of metal-binding site interactions and screening candidate structures are important components of ligand design, we are still missing a vital piece to the puzzle – a way to efficiently generate new host candidates.

Simply put, host design is the process of choosing a set of binding sites and then choosing the connecting geometric structure that ties them together. In many instances, we have sufficient knowledge to identify the number and types of binding sites to complement a given metal, but we need an effective method to identify how to connect these groups together to form an integrated host molecule. At present, we can only generate trial structures by hand with a graphical user interface, an extremely time-consuming process. Often, it is not readily obvious which linkage structures might be best used to connect the binding sites groups.

Drug designers have developed computational approaches to address the inverse of this problem, that is, how to identify a molecular structure (guest) that will bind tightly within the binding site of a protein (host). One approach involves docking fragments into the pocket and then linking them together to yield the potential drug candidate. Software packages that perform these operations require input of the atomic coordinates of a protein binding site and are highly specialized to address protein–organic interactions. Because of this, they are not applicable for the generation of molecular structures designed for metal ion complexation. Our aim was to develop molecular design

algorithms to achieve software that is capable of identifying the best host architctures for a specific metal ion guest. HostDesigner is the result.

The HostDesigner package can be freely downloaded from the website http://hostdesigner.emsl.pnl.gov. It consists of the HostDesigner executable (MacOS or Windows), a linkage structure library (LIBRARY), and one additional file needed to operate the system (CONSTANTS). In addition, several example input files and this manual are included with the download.

HostDesigner does not currently have a graphical user interface. Any text editor can be used to prepare input files. We recommend the use of Chem3D™ software, which outputs Cartesian coordinates and connectivity tables in the format used for input file preparation. The output file can be read with WebLab ViewerLite™ that we use to visualize the host structures. At the time of writing this manual, WebLab ViewerLite™ can be obtained as a free download from the website, http://www.accelrys.com/viewer/. Chem3D™ is a commercial software available through CambridgeSoft, http://www.camsoft.com.

2.0 DESCRIPTION OF THE ALGORITHMS

2.1 The LINKER algorithm.

Host molecules are composed of groups of binding sites. Thus, any multidentate host can be dissected into two or more structural fragments which we can designate as host components. For example, the well known 18-crown-6 macrocycle can be broken down into two triglyme components, three diglyme components, or six dimethylether components. It is possible to define the structure of a complex fragment, in other words, a piece of a host-guest complex, by combining a host component with a guest metal ion. In constructing the complex fragment, the metal ion can be positioned relative to the host component to define a complementary geometry, that is, a geometry that would give the strongest interaction between the binding sites of the host component and the metal ion in an actual complex.

The LINKER algorithm is based on the following assumptions: (1) there is an optimal geometry for the interaction between each binding site of the host and the guest and (2) this optimal geometry is largely independent of the other binding sites that may be present in the host. The LINKER algorithm builds new host structures by connecting two complex fragments to link structures that are taken from a library. Figure 1 shows complex fragments for a number of common donor groups with a variety of host-guest orientations. Although it is true that the M-L distance will increase with increasing coordination number, the preferred orientation of the host component with respect

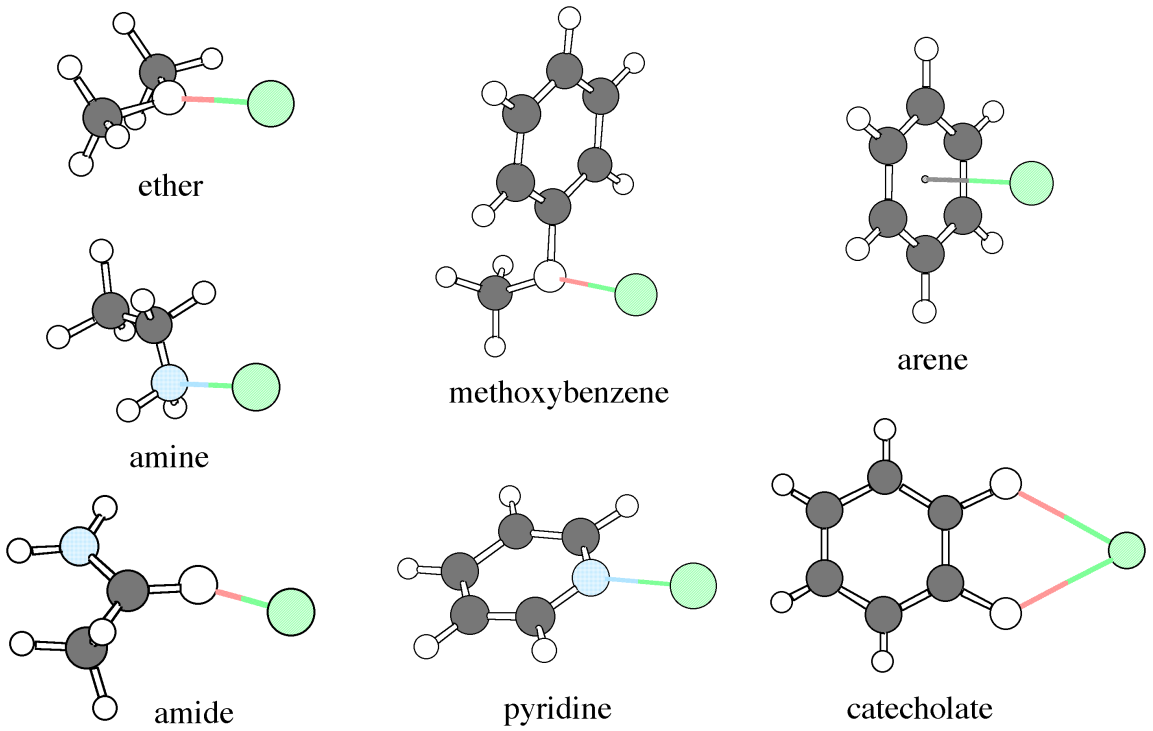


Figure 1. Examples of complex fragments.

to the metal remains largely invariant. Optimal M-L-X angles and M-L-X-X dihedral angles can be deduced from examination of experimental geometries of metal-coordinated ligands or through the careful application of electronic structure calculations. Given a targeted coordination number, the optimal M-L distances can be obtained in the same manner. Thus, it is possible to define an optimal site for metal ion coordination with a given host component.

To use the LINKER algorithm, the user must create an input file for each complex fragment in which the metal ion is in the optimal binding site with respect to the host component. These input files also specify possible points of attachment on each fragment. This is illustrated in Figure 2 where the user has defined two dimethylether lithium components, each with three attachment points. To create a new host structure, LINKER selects a link from the library, forms a bond between the first complex fragment and the link, rotates this bond to a preferred dihedral angle, forms a bond between the the second fragment and the link, and rotates the second bond to a preferred torsion angle. The code then checks the distance between the two metal sites. This distance is used to rank-order the host structure for the degree of fit to the metal. The complex is considered to perfectly complement the metal when the distance is zero, that is, when the two metal atom sites are superimposed.

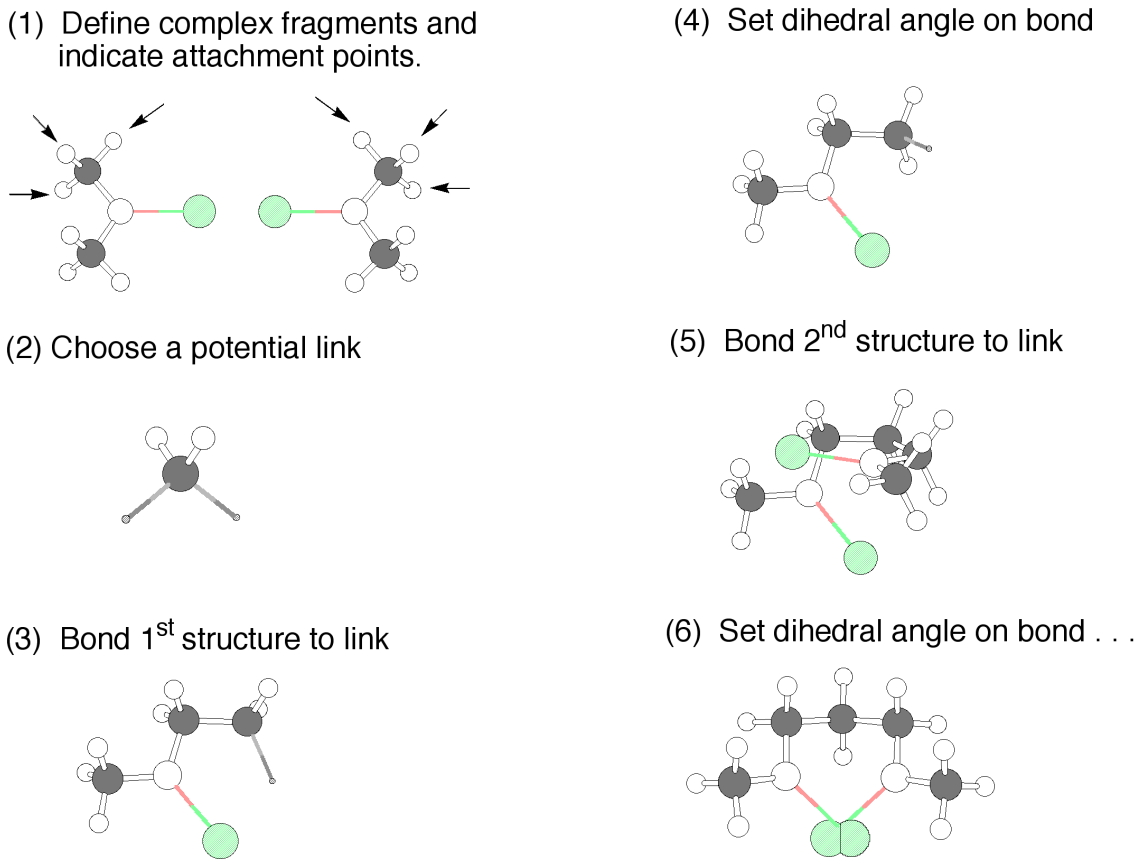


Figure 2. How the LINKER algorithm builds a structure.

Given the possibility of multiple attachment points per component structure and multiple dihedral angles about each bond, it is possible to generate a number of complexes with one link. With the example shown in Figure 2, there are three attachments to each host and there are three possible dihedral angles for each bond giving rise to a total of 81 potential structures. However, when this example is run through the code, only the 9 structures, shown in Figure 3a, are retained. There are two reasons for this. First, subsequent builds may lead to the identical structure or the mirror image of the structure. These degenerate results are rejected. Second, in a number of cases, rotation about the bonds leads to physically unreasonable collision or superposition of generated host atoms (see Figure 3b). When this occurs, such structures are rejected.

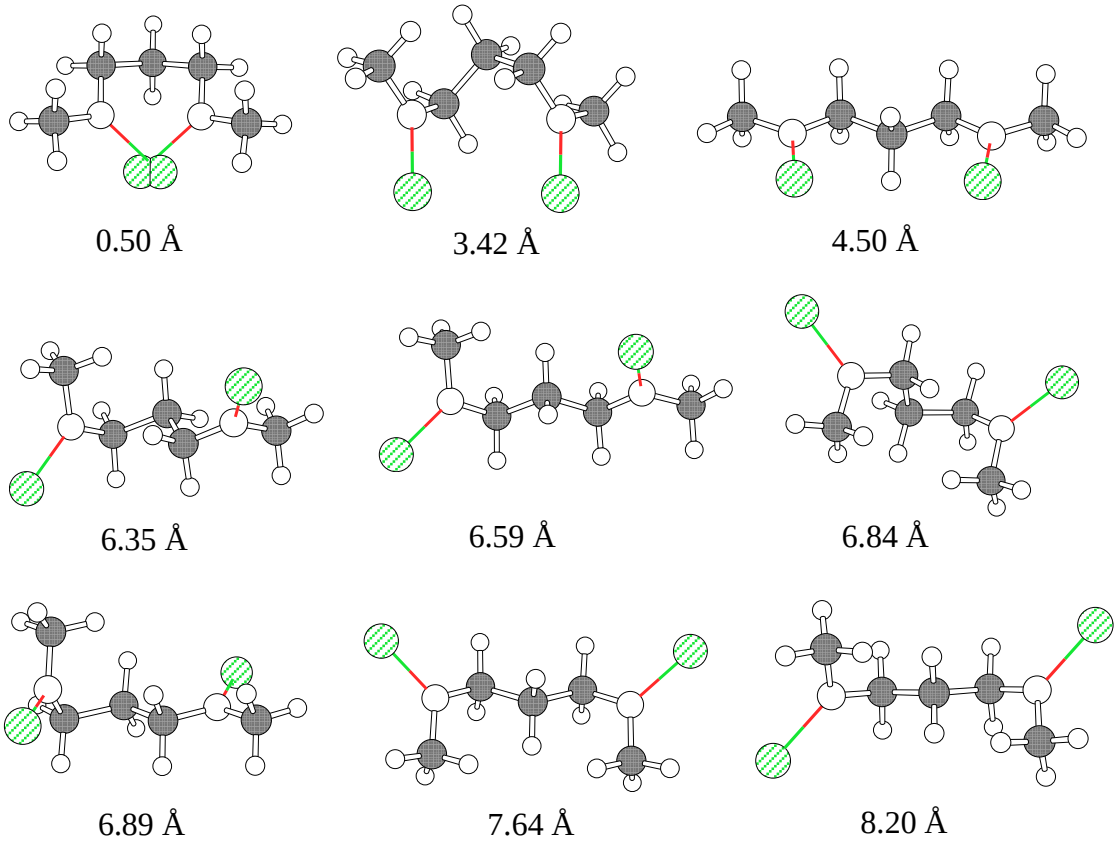


Figure 3a. Host structures resulting from the complex fragments shown in Figure 2.

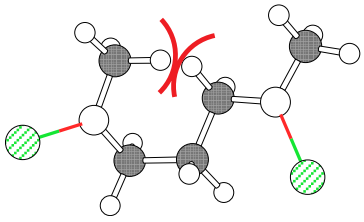


Figure 3b. Example of a structure rejected due to a close H--H contact

During a typical run, LINKER examines every connectivity and bond rotation with multiple link structures. Therefore, large numbers of structures are generated. The resulting hosts are sorted by the metal-metal distance and only the best structures by this criterion are written to the output files. The link library currently contains 8552 potential link structures (see Section 2.3), however the user has some control over which links will be used for building. Searches can be limited by specifying a minimum connectivity, a maximum connectivity, the valence of the bonding atoms, and the maximum conformational energy (when a link derives from a structure that has more than one conformer). Section 3.0 provides further detail on how to control this application.

The choice of the dihedral angles for the two bonds that are formed to the link is a critical aspect of the LINKER algorithm. At this time, LINKER only treats six C-C bond types: C(alkane)-C(alkane), C(alkane)-C(alkene), C(alkane)-C(arene), C(alkene)-C(alkene), C(alkene)-C(arene), and C(arene)-C(arene). Thus, the input structures made by the user must contain at least one carbon atom (alkane, alkene, or arene) defined as an attachment point. To assign the dihedral angles to each C-C bond, the code considers both the type of carbon atoms and the degree of substitution on each of the carbon atoms. The values that are assigned are based on examination of MM3 potential energy surfaces for rotation about the 66 possible types of bonds formed by all combinations of the 11 groups shown in Figure 4.

Note that in addition to the types of bonds that can be formed, there are other limitations to the LINKER algorithm. During the early stages of development, it was realized that it was not possible to obtain a macrocyclic structure using LINKER. The reason is illustrated in Figure 5. When two 1,3-propane diamine chelates are directly connected and the bond is rotated to an optimal dihedral angle, the code generates a complementary host structure that corresponds to a stable conformation of the known cyclam macrocycle. This structure was rejected, however, because the terminal methyl groups are superimposed. This problem led to the development of an algorithm to detect and allow the formation of macrocyclic structures. At this time, the algorithm only allows the formation of bonds between two terminal methyl groups. The user can either invoke or disable this algorithm while running HostDesigner in the LINKER mode (see Section 3.1 and 3.2).

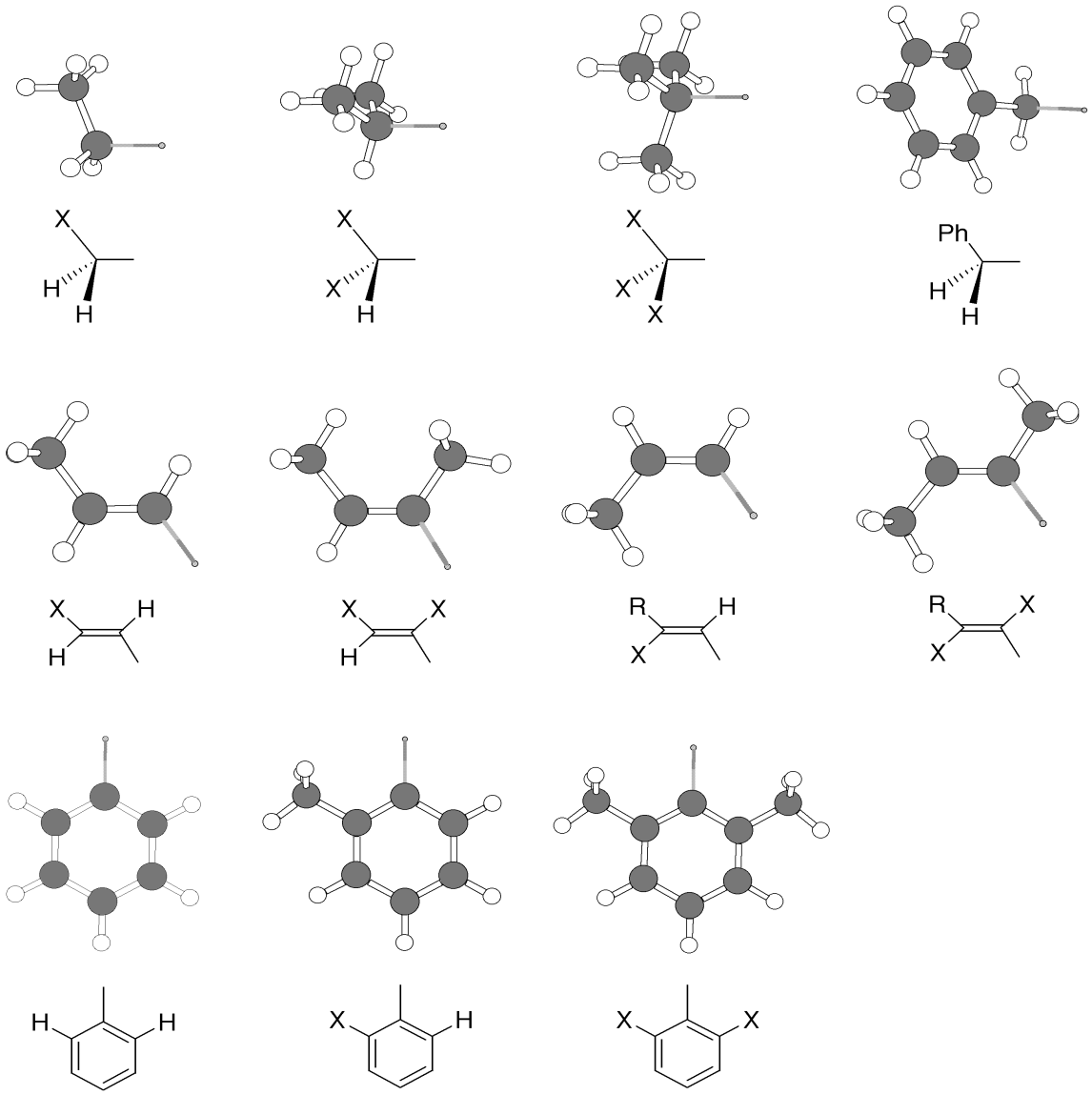


Figure 4. Groups used to assign rotational minima about C-C bonds.

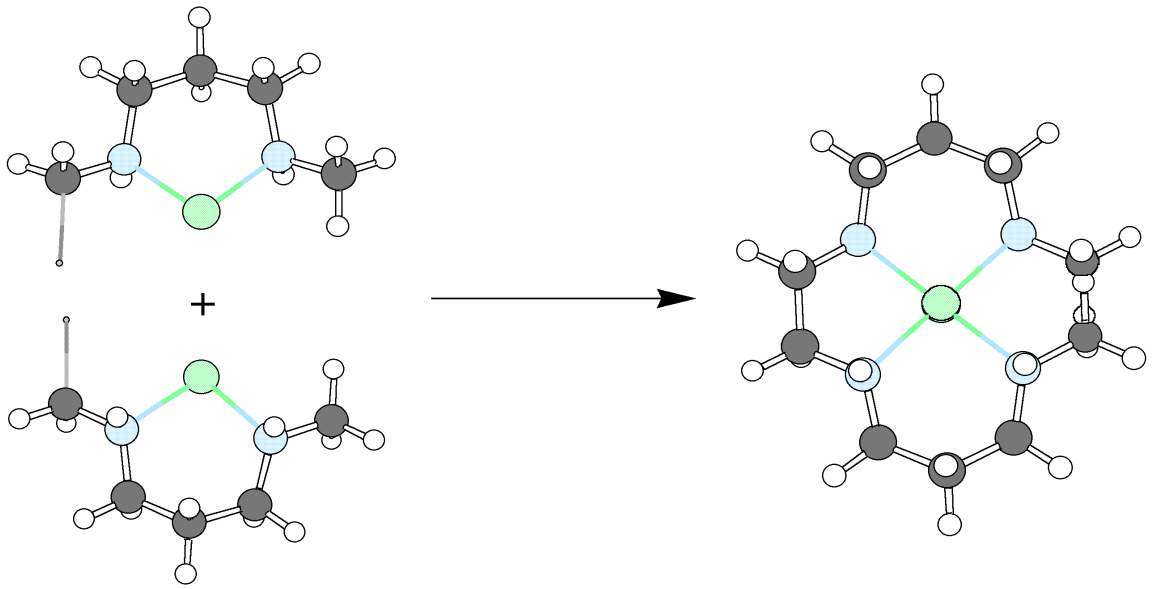


Figure 5. The combination of two 1,3-propanediamine chelate rings yields a macrocyclic structure that would be rejected because of the superposition of the terminal methyl groups.

Complementarity, as defined by metal-metal distance, addresses optimal M-L distances, M-L-X angles, and M-L-X-X dihedral angles, but does not adress L-M-L angles. Fortunately, this definition is appropriate for the Group 1A and 2A metals, the lanthanides, most actinides, and many transition metals. In cases where there is marked directionality at the metal center, for example, square planar complexes found with Cu(II), Pt(II), and Pd(II) and the polyhedra associated with oxo species like TcO^{3+} and UO_2^{2+}, the default LINKER algorithm may yield hits with ligands coordinated at positions far from the optimal shape for such metals. In these cases, it may be useful to apply a restriction to the metal topology.

There are six keywords allowed for restricting the metal topology, or the shape that the ligands make about the metal. We emphasize that this topology screening process can only reject hits it finds to be outside of tolerance of the specified shape; it does not generate any additional structures. The default metal topology keyword is 'NONE', which ignores the orientation of the ligands about the metal and is recommended for non-directional metals such as alkali, alkiline earths, f-block metals, most early transition metals., and any other non-directional metal center. The keywords to activate a shape-based screening are: TETR (tetrahedral); SQPL (square planar); TBPY (trigonal bipyramidal); SQPY (square pyramidal; a 96° angle to the apex was used as the ideal); and OCTA(octahedral).

The keyword can be applied to partial occupations of a target shape. For example, a monodentate and a bindentate complex fragment combined with the OCTA keyword on would screen out all potential new tridentate complexes which were not approximately a face of an octahedron (fac) or meridianal around an octahedron (mer). The program will try to map to all the possible connections to an ideal shape with exception that in

bidentate cases *trans*-orientations are rejected. In cases where binding sites must be located at specific positions on a metal ion polyhedron, the OVERLAY algorithm may also be of use. (see Section 2.2).

The authors would like to caution the user from over interpreting the results obtained from the LINKER algorithm. This algorithm is designed to examine large numbers of potential structures in a short period of time and to identify connectivities and conformations that yield complementary architectures for metal ion complexation. Every effort has been made to provide accurate link structures and dihedral angle assignments. However, please recognize that small changes in individual structural parameters, which often do not cost much in terms of energy, can have a significant impact on the resulting structure. Because the code is connecting rigid components and using generic dihedral angles, the ranking of the structures based on the distance between metal sites should be regarded as somewhat crude. When these distances differ by angstroms, then the difference between convergent and divergent binding sites is clear (see, for example, Figure 3). However, when these distances differ by less than an angstrom, the ranking becomes more uncertain. To obtain a more accurate ranking, the list of structures initially generated by LINKER should be subjected to further analysis using more expensive methods such as molecular mechanics or electronic structure calculations.

Finally, it should be noted that the LINKER algorithm ranks structures only in terms of their complementarity for the metal ion. In other words, LINKER does not consider the fact that a highly complementary structure may also be a very high energy conformer for the host. It is currently left to the user to perform conformational analysis on promising host architectures to determine their degree of preorganization.

2.2 The OVERLAY algorithm.

The OVERLAY algorithm, which contains a completely different approach than that used in the LINKER algorithm, is for use in cases where the user desires to add structure to an existing host architecture. Two examples of such applications are shown in Figure 6. In the first example (left), a potential macrocyclic architecture has been identified after several passes through the LINKER routine. Is it possible to find a connecting structure that bridges the two terminal carbons? In the second example (right), U(IV) forms a stable tetrakis-catecholate complex with a dodecahedral geometry. Is it possible to find a connecting structure to that will bridge two catecholates at the ortho carbons of the catecholate fragments?

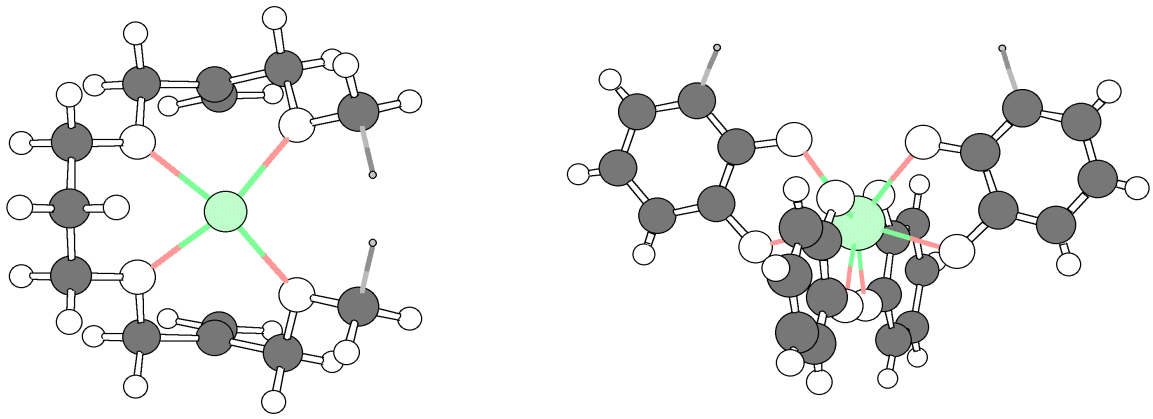


Figure 6. Examples of cases where the OVERLAY algorithm would be used.

To use the OVERLAY algorithm, the user makes an input file that defines the structure of a complex fragment and specifies pairs of C-H bonds that can be replaced with C-C bonds. To build a new structure, the code takes a link from the library and tries to overlay the link's bonding vectors with the hosts bonding vectors. Prior to the overlay, the lengths of the bonding vectors on both the host and the link are adjusted to the proper length for the type of bond that would be formed, based on the atom types of the atoms to be bonded. After the overlay, the degree of fit is indicated by the root mean squared displacement of four points on the host with four points on the link, where in each case the four points correspond the ends of the bonding vectors. OVERLAY will try multiple links and rank them by the degree of fit. As with the LINKER routine, the user has some control over which links will be used. Searches can be limited by specifying a minimum connectivity, a maximum connectivity, the valence of the bonding atoms, and the maximum conformational energy (when a link derives from a structure that has more than one conformer). Sections 3.1 and 3.3 provide further detail on how to control this application.

2.3 The Linkage Library

(a) Library development. A linkage library file named LIBRARY is provided with the HostDesigner software. This section of the manual describes how it was developed. In building the initial library, it was decided 1) to limit the entries to molecules containing hydrogen and up to six carbon atoms, 2) to limit carbon hybridization to C(sp^2) and C(sp^3), and 3) to exclude three- and four-membered rings. With these limitations, a systematic evaluation yielded a total of the 81 connectivities shown in Figure 7a. Subsequently, 19 additional connectivities representing rigid bi- and tri-cyclic systems (see Figure 7b) have been added. These connectivities were used to create a total of 8552 links.

The process of creating a link is summarized in Figure 8. The process starts by selecting a hydrocarbon molecule. 1) The molecule is conformer searched. For each link, 2) a conformer is selected and 3) a pair of hydrogens are replaced by methyl groups and the structure is reoptimized with the MM3 program. When the link structure is used by Hostdesigner, the complex fragments will add steric bulk at these sites; by optimizing with methyl groups at these sites the link structure is pretreated for future attachment to carbon substituents. 5) After optimization, the methyl groups are then replaced by dummy atoms with C-dummy bond lengths of 1.0 Å. 6) The new link is checked for degeneracy, that is, to make sure that it neither identical to or the mirror image of an existing link. It is then added to the library. This process is repeated for every possible pair of hydrogens on the molecule and for every conformer of the molecule.

Given the above process, note that the library does not contain enantiomers. However, if a link is chiral, then both the LINKER and OVERLAY algorithms can consider the chiral image of the link during the building process. This option is activated by the user (see Section 3.0).

Initially, all new links were simply entered into the library in order of increasing molecular weight. However, as the size of the library increased, we encountered a problem with this approach. This problem has to do with the fact that many links have very similar geometry. For example, the 1,1-ethane link has bonding vectors that are almost the same as the 1,1-propane, 1,1-butane, 1,1-pentane, and 1,1-hexane links. Thus, if a 1,1-alkane linkage gives the best result, the code will generate an entire family of 1,1-alkane linked structures that will dominate the output. To overcome this problem, we adopted a method of grouping link structures into families based on geometric features of the bonding vectors. At the same time, we developed a classification scheme to organize the library with respect to the connectivity, the hybridization of the bonding atoms, and the molecular weight of the link. Now the library is organized by class and each link belongs to a class. During a run, the user has the option to examine every link in the library or to limit examination to the first member of each class. The current library of 8552 links represent a total of 2950 classes.

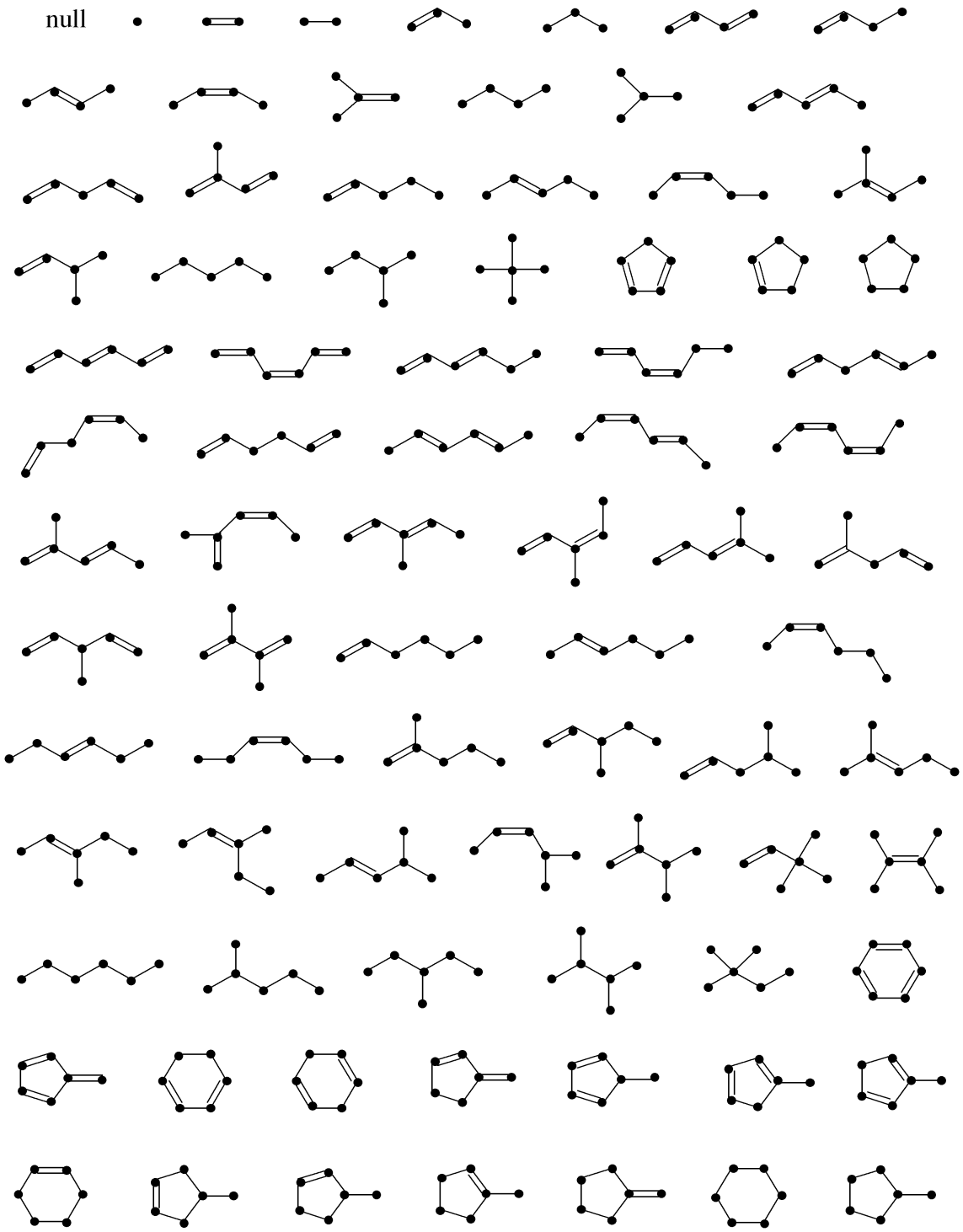


Figure 7a. Connectivities for 0 to 6 carbons used to build the initial link library.

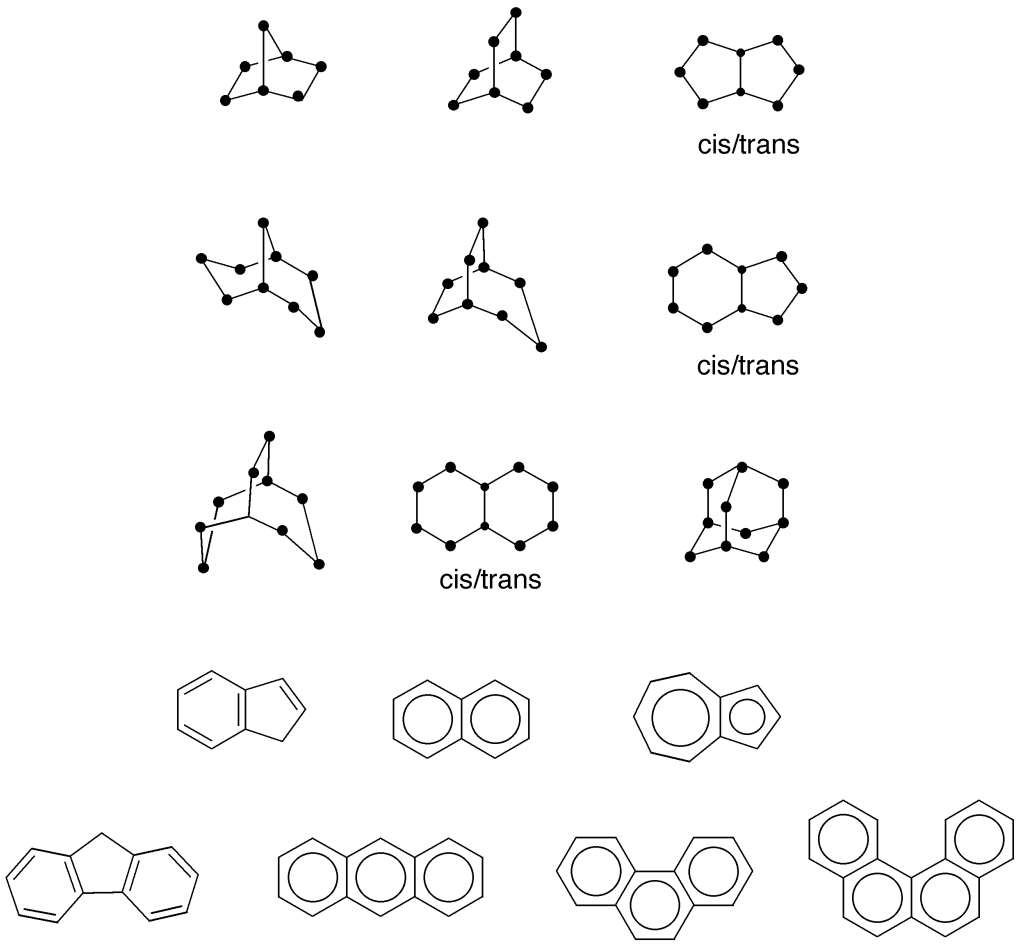
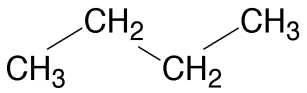
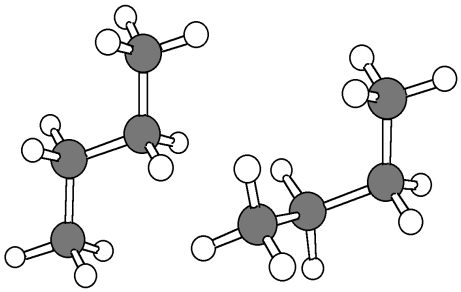


Figure 7b. Additional rigid bi- and tri-cyclic connectivities added to the link library.

(1) Select a hydrocarbon. Here we are using n-butane.

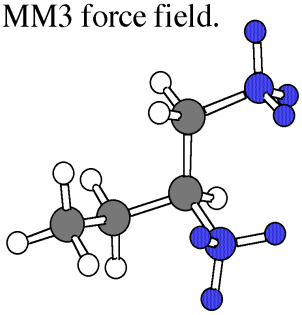


(2) Search to identify all stable conformers. Here there are two.

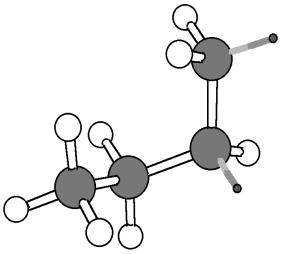


(3) For each conformer, identify all possible hydrogen pairs. With 10 hydrogens, these conformers each have 45 hydrogen pairs, in other words, n-butane could yield as many as 90 linkages.

(4) For each hydrogen pair, replace hydrogens with methyl groups and optimize the structure with the MM3 force field.



(5) Replace the methyl groups with dummy atoms and set the dummy atom to carbon distances to 1.0 Å.



(6) Check for degeneracy, that is, the link is neither identical or the mirror image of an existing link.

(7) Add the new link to the library.

Figure 8. A summary of the link making process.

(b) Format of a link entry . The link library consists of a sequential series of entries. Each entry contains information that completely describes a link. Below is an example of the entry for a link made from butane. This file is read in a free format fashion which means that it is not critical how many spaces are left between each entry on a line. The blue numbers on the right of this sample library entry are line numbers for refererence in this manual, and are not used in practice.

```
LINK     C( 4),H( 8)     t_butane                                          1
   13    1   C     3                                                       2
   14    1   C     3                                                       3
    1   .false.  .true.     0.000000    1.146900                          4
    0.000000  144.807545  144.807545    0.000000                          5
   14                                                                      6
    1   C    -0.934460    0.028700   -0.365680    1    2   13   14    5   7
    2   C     0.430390    0.511920    0.168490    1    1    3    6    7   8
    3   C     1.625320   -0.331380   -0.306550    1    2    4    8    9   9
    4   C     2.967210    0.245020    0.164330    1    3   10   11   12  10
    5   H    -0.888010   -0.008600   -1.481300    5    1                 11
    6   H     0.408570    0.526720    1.282370    5    2                 12
    7   H     0.590010    1.566140   -0.156830    5    2                 13
    8   H     1.621860   -0.389820   -1.419560    5    3                 14
    9   H     1.529560   -1.374220    0.072210    5    3                 15
   10   H     3.821230   -0.378530   -0.183530    5    4                 16
   11   H     3.023760    0.294120    1.274740    5    4                 17
   12   H     3.129730    1.274910   -0.224880    5    4                 18
   13   X    -1.651036    0.675600   -0.104836    0    1                 19
   14   X    -1.165726   -0.883163   -0.026532    0    1                 20
```

Line 1: Always begins with LINK. Other information on this line is not read by the code. Here we have included the molecular formula and a name.

Line 2 and 3: Each link has two attachments (in this case, 'atoms' 13 and 14.) These two lines specify the attachment points. There are four variables per line: the serial number of the dummy atom, the serial number of the bonding atom, the atom label of the bonding atom, and the hybridization of the bonding atom (2 = sp^2, 3 = sp^3).

Line 4: Contains five variables: the connectivity, attachment symmetry, chirality, relative energy of the conformer, and relative energy of the methylated link.

The connectivity is an integer between 0 and 999. It is the number of atoms between the two dummy atoms.

The attachment symmetry is a logical variable with possible values of .true. or .false.. The donor group structures can be attached to the link in two ways. If swapping the attachment results in the same molecule, then the attachment symmetry is true. Otherwise, the attachment symmetry is false.

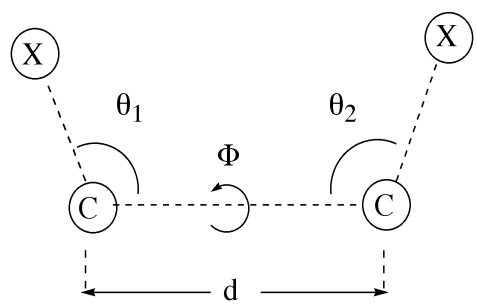
The chirality is a logical variable with possible values of .true. or .false.. If the mirror image of the link is not superimposable on the link, then chirality is true. Otherwise, it is false.

The relative energy of the conformer refers to the hydrocarbon used to make the link. The value is zero if there is only one conformer or if this link came from the

lowest energy conformer. This variable can be used to eliminate high energy link conformations from the search.

The relative energy of the methylated link was obtained during link construction when optimized with two methyl groups present. This variable is not currently used by the code and any value can be entered.

Line 5: Contains four variables that describe the geometry of the bonding vectors: distance (d), angle1 (θ_1), angle2 (θ_2), and dihedral angle, (Φ). These values are used to classify the link and also used in the OVERLAY algorithm to decide whether to attempt an overlay operation. The parameters are defined as shown in the scheme below:



where d is the distance between the two carbon atoms, θ_1 is the angle between the first C-X vector and the C-C vector, θ_2 is the angle between the second C-X vector and the C-C vector, and Φ is the dihedral angle between the two carbon-dummy vectors about the carbon-carbon axis. When both dummy atoms are attached to the same carbon, as in this example, then $\theta_1 = \theta_2 = 90° + 1/2$ the angle between the two C-X vectors and $\Phi = 0°$. In the existing library, the molecule is numbered such that θ_1 is always $\leq \theta_2$. In addition, the chirality of the molecule is chosen such that Φ is always in the range of 0 and 180°.

Line 6: The number of atoms in the link including the two dummy atoms.

Lines 7 to end: There is one line per atom. Each line contains the following informaton: The serial number, the atom label, x, y, and z coordinates, the MM3 atom type number, and a connectivity list by serial number. The atom label is a two character string. It is assigned a value of X for the dummy atoms. The MM3 atom type numbers used in the current library are limited to 1 (sp^3 carbon), 2 (sp^2 carbon), 50 (arene carbon), 5 (hydrogen) and 0 (dummy atom). The connectivity list is simply a list of all serial numbers to which this atom is attached. The connectivity can be up to twelve atoms. Note that the serial order of the atoms is important. The carbons always go first, the hydrogens always go next, and the dummy atoms always go last.

(c) How to add a new link to the library. The user can create additional links to expand the existing library or they can define their own personal link library. Assuming that the user has created a new link and formatted a link entry as described above, the entry should be added to the existing library as follows.

The library is organized by class and each class section begins with a header line. Before adding a link to the library, it is necessary to identify the class to which the new link belongs. Classes are defined by the connectivity, in other words, the number of carbon atoms that would be between two groups connected to this link, and by the hybridization of the bonding carbons. Classes are arranged first in order of increasing connectivity and next in order of increasing hybridization. If the new link does not belong to an existing class, then you must create a new class header line and insert it in the proper position in the library.

An example of a class header line is given below. All of the information on this line is read in fixed format and the line must be written exactly as shown, in other words, spaces matter. Connectivity can range from 0 to 99. Hybridization is either C(sp^2),C(sp^2), or C(sp^2),C(sp^3), or C(sp^3),C(sp^3).

```
CLASS  connectivity= 1  C(sp3),C(sp3)  type(  1)
```

Each class is further subdivided into types, where a type is defined by the geometry of the bonding vectors as defined above. In the construction of the current library, a link belongs to an existing type if the difference between the geometric parameters of the first link in this type and the new link are less than 0.1 Å for distance, 3° for angles, and 5° for dihedrals. If any one of the differences exceeds these limits, then a new type is created and the link is entered as the first member of this type.

3.0 HOW TO DRIVE HOSTDESIGNER

3.1 Installation

Before you run HostDesigner, first set up a directory that contains the HostDesigner executable and the two files provided with the code named 'LIBRARY' and 'CONSTANTS'.

3.2 The File Named 'control'

To run HostDesigner, you must make some input files and place them in the directory that contains the code. One of these input files must be named either 'control' or 'control.txt'. It is the file that tells HostDesigner what to do.

A 'control' file is composed of a list of keywords. These keywords are entered on lines that are 80 characters in length. You may put as many keywords on one line that will fit within this length. If more than one line is used, then you must specify that another line will be read by placing the keyword 'AND' in the preceding line.

Here is an example of acceptable format for the 'control' file:

```
keyword1  keyword2  keyword3  keyword4  keyword5
```

And this is also an acceptable way to format the same information:

```
keyword1  keyword2  keyword3  AND
keyword4  keyword5
```

Here is an example of a simple 'control' file for a LINK run:

```
LINK  hosta=Ni_lig1  hostb=Ni_lig2
```

Here is 'control' file for a LINK run in which more options have been selected:

```
LINK hosta=Ni_lig1 hostb=Ni_lig2 mirroraoff mirrorboff AND
minconn=1 maxconn=2 maxconfe=1.0d0 out=dmeli_2_octa numkeep=100 AND
metshape=OCTA
```

Keywords can be given in any order. Keywords must be separated from one another by at least one space. Some keywords are used specify additional numeric or string input. In such cases, no spaces may be present within the keyword. For example, '`numkeep = 100`' will result in an error while '`numkeep=100`' will not. Note also that keywords are case sensitive. The following are a list of keywords that are allowed by HostDesigner.

Mandatory Keyword:

one, and only one, of these two keywords must be present in the ‘control’ file.

LINK Tells the code to do a LINKER run

OVER Tells the code to do an OVERLAY run

Optional Keywords:

hosta=name Specifies the name of the file HostDesigner reads for the host in OVERLAY, or the name of the file containing the first complex fragment in LINKER (See Sections 3.2 and 3.3). The name may be set to 20 characters in length. If no name is specified, then the default filename is ‘hosta’.

hostb=name Specifies the name of the file containing the second complex fragment in LINKER (See Section 3.2). The name may be set to 20 characters in length. If no name is specified, then the default filename is ‘hostb’.

linklib=name Specifies the location of the link library. The name may be no more than 60 characters in length. If no name is specified, then the default filename is ‘LIBRARY’.

mirroraoff By default, LINKER uses the ‘hosta’ structure and its mirror image. This keyword turns this feature off and uses only the structure specified in ‘hosta’. This is not recommended for use with chiral or prochiral complex fragments.

mirrorboff By default, LINKER uses the ‘hostb’ structure and its mirror image. This keyword turns this feature off and uses only the structure specified in ‘hostb’. This is not recommended for use with chiral or prochiral complex fragments.

macro Turns on the macrocycle feature of LINKER.

metshape=name Reads a 4 character string that may be set equal to NONE, TETR, SQPL, TBPY, SQPY, or OCTA. The default setting is NONE. If set to a value other than NONE, host structures are rejected when the binding sites do not fit the specified topology.

useclassoff By default, only one member of each link class is used. This keyword forces HostDesigner to use every structure in the link library.

minconn=# An integer equal to the minimum number of atoms in the minimal path through the link, this defaults to 0. If set higher, the shorter path structures in the link library will be skipped.

maxconn=# An integer equal to the maximum number of atoms in the minimal path through the link, this defaults to 9. If set lower, the longer path structures in the link library will be skipped.

maxconfe=# A real number which defaults to 100.0. A potential link conformer in the link library will be skipped if it has an MM3 energy higher than maxconfe kcal/mol with respect to that isomer's global minimum.

out=name The out string is used as the prefix for the output files, which will be named '<name>.summ' and '<name>.xyz' (See Section 3.4). The default setting is 'out'.

numkeep=# The maximum number of structures to be printed in '<name>.xyz'. The default is set to the maximum number of 100.

AND Used when more than one line is needed in the control file. Instructs the code to continue reading input from the next line. If the line being read does not contain an 'AND' keyword, then all subsequent lines are ignored.

Here is an example of a 'control' file that presents the minimal amount of input:

```
OVER
```

This control file tells the code to use the OVERLAY algorithm. It will look for the default names of the other input files, in other words, the link library must be named LIBRARY and the complex fragment must be named 'hosta'. Output files will be named 'out.xyz' and 'out.summ'. All other options will be at their default settings.

Here is an example of a 'control' file that uses every keyword:

```
LINK hosta=one hostb=two linklib=short mirroraoff mirrorboff AND
macro metshape=OCTA useclassoff minconn=1 maxconn=3 maxconfe=0.0 AND
out=three numkeep=50
```

This control file tells the code the use the LINKER algorithm with input files named 'one' and 'two', links from a library named 'short', not to consider mirror images, use the macrocycle feature, screen for octahedra, consider all links in the library, screen out links with connectivities < 1 or > 3, use only links derived from minimum energy conformers, output files will be 'three.xyz' and 'three.summ', and 50 structures will be printed to 'three.xyz'

3.2 Other Input Files for the LINKER mode.

In addition to the 'control' file, the LINKER mode requires the user to prepare input files that define the structure of the complex fragments that will be connected to the link structures. The two complex fragment files can be identical or two different complex fragments can be used. Each complex fragment must contain at least one binding site coordinated to the metal ion. However, the complex fragment may contain up to twelve coordinated binding sites. In addition, in order to specify an attachment point, the complex fragment must contain at least one C-H bond involving either an sp^3, sp^2, or arene carbon atom. The following is an example of the complex fragment file for the dimethylether lithium complex shown in Figure 2 above. All lines are free format so that it does not matter how many spaces are left between the variables.

```
Lithium dimethylether, M-O = 2.26 Å                                      1
 10                                                                      2
  C    1   -1.177597   -1.175812   -0.004669   1    2    4    5    6     3
  O    2   -0.379028   -0.000275   -0.004684   6    1    3   10          4
  C    3   -1.173340    1.178436   -0.004654   1    2    7    8    9     5
  H    4   -0.474350   -2.040863   -0.013016   5    1                    6
  H    5   -1.744690   -1.178024    0.955856   5    1                    7
  H    6   -1.757370   -1.169617   -0.958374   5    1                    8
  H    7   -1.735809    1.185837    0.958420   5    3                    9
  H    8   -1.757660    1.171387   -0.955414   5    3                   10
  H    9   -0.466919    2.040848   -0.019073   5    3                   11
 Li   10    1.880707   -0.001373   -0.036957 125    2                   12
  2                                                                      13
  4   C  0                                                               14
  5   C  0                                                               15
```

Line 1: Title for the donor group that can be up to 60 characters long.

Line 2: An integer, n, equal to the number of atoms in the structure.

Lines 3 to 2 + n: Each line describes an atom in the structure and contains the following information: the atom label, the serial number, the x, y, and z coordinates, the MM3 atom type, and the connectivity list. It is important that atoms be in the serial order with the heavy host atoms first, the hydrogens next, and the metal ion guest last. In addition, it is important that the atoms be sequentially numbered from 1 to n. This section of the file can easily be prepared using the Chem3D™ software by saving the file in 'cartesian 1' format with serial numbers, atom type numbers, and connectivity by serial number activated.

Line 3+n: An integer, na, giving the number of attachment points to the structure. Currently the maximum number of attachments is 100.

Line 4+n to 3+n+na: The attachment sites. Each attachment site is specified by three variables. These are the serial number of the hydrogen to be lost, an atom label indicating the type of atom that the link must have in order to bind, and an integer indicating the hybridization of the atom that the link must have in order to bind. At this time all the links in the library bind with carbon atoms. Thus the atom label should either be C or XX (which indicates any atom label.) The final number, the hybridization, is defined as follows: 0 = any hybridization, 2 = sp^2 hybridization, or 3 = sp^3 hybridization.

Line 4+n+na: (if the option macrocycle is turned on in the control file) An integer equal to the serial number of the terminal methyl carbon which may form a macrocycle.

3.3 Other Input Files for the OVERLAY mode.

In addition to the ‘control’ file, OVERLAY requires the user to prepare an input file that defines the structure of a complex fragment. This structure must contain pairs of binding sites to which the links will be attached. These sites must be of C-H bonds involving either alkane, alkene, or arene carbon atoms. The following is an example of a complex fragment made from the lithium tetradentate ether complex in Figure 6 above.

```
tetradentate ether podand with lithium
 43
  C    1    2.395416   -0.164734    3.161545    1    2   25   26   27   1
  O    2    2.057663   -0.361847    1.791534    6    1    3   43        2
  C    3    3.113632   -0.974075    1.057251    1    2    4   21   22   3
  C    4    2.674850   -1.130783   -0.377457    2    3    5    6        4
  C    5    2.343353   -2.327927   -0.887054    2    4   19   20        5
  C    6    2.600372    0.129227   -1.203751    1    4    7   23   24   6
  O    7    1.459473    0.875443   -0.790359    6    6    8   43        7
  C    8    1.289703    2.077225   -1.536667    1    7    9   28   29   8
  C    9    0.024597    2.754669   -1.072205    2    8   10   11        9
  C   10    0.052734    3.879349   -0.339691    2    9   30   31       10
  C   11   -1.273315    2.088425   -1.455261    1    9   12   32   33  11
  O   12   -1.405731    0.887848   -0.699341    6   11   13   43       12
  C   13   -2.576996    0.151703   -1.039261    1   12   14   34   35  13
  C   14   -2.609787   -1.107895   -0.209488    2   13   15   18       14
  C   15   -2.955215   -0.947784    1.250076    1   14   16   38   39  15
  O   16   -1.849487   -0.344940    1.915680    6   15   17   43       16
  C   17   -2.097855   -0.145264    3.304306    1   16   40   41   42  17
  C   18   -2.321700   -2.307693   -0.738785    2   14   36   37       18
  H   19    2.381836   -3.239563   -0.269714    5    5                 19
  H   20    2.005920   -2.416809   -1.931976    5    5                 20
  H   21    3.357880   -1.967026    1.506943    5    3                 21
  H   22    4.036606   -0.347290    1.108414    5    3                 22
  H   23    2.515915   -0.117081   -2.289993    5    6                 23
  H   24    3.529556    0.733978   -1.068512    5    6                 24
  H   25    2.598343   -1.175278    3.590515    5    1                 25
  H   26    3.282593    0.506653    3.254745    5    1                 26
  H   27    1.540451    0.310806    3.683853    5    1                 27
  H   28    2.175644    2.740814   -1.385712    5    8                 28
  H   29    1.221237    1.851593   -2.628250    5    8                 29
  H   30    1.013367    4.339005   -0.057587    5   10                 30
  H   31   -0.877869    4.355515    0.008057    5   10                 31
  H   32   -1.276215    1.862427   -2.548920    5   11                 32
  H   33   -2.142075    2.759552   -1.248566    5   11                 33
  H   34   -3.490479    0.764420   -0.845444    5   13                 34
  H   35   -2.563843   -0.095000   -2.128571    5   13                 35
  H   36   -2.052017   -2.399185   -1.803040    5   18                 36
  H   37   -2.328812   -3.219162   -0.120041    5   18                 37
  H   38   -3.179000   -1.938705    1.714706    5   15                 38
  H   39   -3.867584   -0.313004    1.359558    5   15                 39
  H   40   -2.971558    0.533722    3.453506    5   17                 40
  H   41   -2.281906   -1.154129    3.745605    5   17                 41
  H   42   -1.207397    0.322693    3.771118    5   17                 42
 Li   43    0.075745    0.235489    0.882080  125    2    7   12   16  43
  2                                                                     44
 25   C   0                                                             45
 41   C   0                                                             46
```

Line 1: Title for the donor group that can be up to 60 characters long.

Line 2: An integer, n, giving the number of atoms in the structure.

Lines 3 to 2+n: Each line describes an atom in the structure and contains the following information: the atom label, the serial number, the x, y, and z coordinates, the MM3 atom type, and the connectivity list. It is important that atoms be in the serial order with the heavy host atoms first, the hydrogens next, and the metal ion guest last. In addition, it is important that the atoms be sequentially numbered from 1 to n. This section of the file can easily be prepared using the Chem3D™ software by saving the file in 'cartesian 1' format with serial numbers, atom type numbers, and connectivity by serial number activated.

Line 3+n: An integer, na, which is the number of attachments to the structure. Attachments are used in pairs in overlay, so this must be an even number. A maximum of 100 attachments, that is a maximum of 50 pairs, can be specified.

Line 4+n to 3+n+na: The attachment sites. Each attachment site is specified by three variables. These are the serial number of the hydrogen to be lost, an atom label indicating the type of atom that the link must have in order to bind, and an integer indicating the hybridization of the atom that the link must have in order to bind. At this time all the links in the library bind with carbon atoms, thus the atom label should either be C or XX(a wildcard which indicates any atom label.) The final number, the hybridization, is defined as follows: 0 = any hybridization, 2 = sp^2 hybridization, or 3 = sp^3 hybridization. The program will sequentially attempt to overlay to each pair of atoms specified.

3.4 Descripton of the Output Files.

The output from HostDesigner consists of a series of structures, ordered with respect to their degree of fit. At this time the code is limited to output a maximum of 100 structures. In either mode of operation, LINKER or OVERLAY, the code will write two output files with suffixes appended to the root name specified in the control file.

One of these files, '<name>.summ' is a summary that tells the user how many links were examined, how many structures were built, how many structures were retained, and how long it took to do the run. It also reproduces the control file and prints the title line from each of 'hosta' and 'hostb' files.

The other file, '<name>.xyz' contains a list of Cartesian coordinates for the host structures that were generated. The format for a structure entry in this file is illustrated in the example shown on the next page. In this case, structures are presented in order of increasing metal-metal distance. The format of an OVERLAY output is identical, but in this case the structures are presented in order of decreasing degree of superposition of the bonding vectors.

The '<name>.xyz' file can be read by the freeware WebLab ViewerLite™, http://www.accelrys.com/viewer/. When the file is opened, all of the molecules appear in a jumble on the screen. To view them one at a time, follow this procedure. First under the pull-down menu entitled window, select hierarchy view. Then under the same pull down window, select tile vertical. You should now have two windows next to each other. One on the right contains the jumble of molecules and one on the left contains a list of hits. The program offers the facility to view each individual molecule from the list. On a Macintosh computer, this is done by selecting a hit in the hierarchy view window to highlight it, and then pressing the 'apple' and 'j' keys simultaneously.

Example format for a single entry in a ‘<name>.xyz’ file:

```
34                                                                  1
Hit(__1),_RMSD(__0.2803)                                            2
C      -4.81297   1.60888  -2.38351                                 3
O      -4.92203   1.44246  -0.97638                                 4
C      -6.19171   1.85867  -0.49163                                 5
H      -5.56556   0.92782  -2.84608                                 6
H      -4.89942   2.70333  -2.58462                                 7
H      -6.94749   1.17459  -0.94415                                 8
H      -6.27474   2.95063  -0.70642                                 9
H      -6.17889   1.70582   0.61266                                10
Li     -3.26606   0.64608   0.33930                                11
C      -3.41489   1.14153  -2.77294                                12
C      -3.20979  -0.36580  -2.52248                                13
C      -1.82344  -0.81211  -2.91283                                14
C      -0.81486  -1.14926  -2.09384                                15
C      -0.82516  -1.16293  -0.58612                                16
C       0.00000   0.00000   0.00000                                17
H      -2.65925   1.72089  -2.19287                                18
H      -3.23808   1.36987  -3.84973                                19
H      -3.95399  -0.94977  -3.11084                                20
H      -3.41482  -0.59175  -1.45413                                21
H      -1.62864  -0.83093  -4.00049                                22
H       0.15020  -1.42556  -2.55604                                23
H      -1.86359  -1.11031  -0.19449                                24
H      -0.40992  -2.13448  -0.23300                                25
H       1.05070  -0.06377  -0.36716                                26
H      -0.41086   0.96929  -0.36713                                27
C       0.00000   0.00000   1.52470                                28
O      -1.33900   0.24473   1.93309                                29
C      -1.46600   0.25749   3.34872                                30
H       0.60839   0.86149   1.88789                                31
H       0.26443  -1.02165   1.88808                                32
H      -0.86664   1.12321   3.71647                                33
H      -1.18934  -0.76337   3.70472                                34
H      -2.54284   0.44045   3.57218                                35
Li     -3.06638   0.54251   0.50659                                36
```

Line 1: An integer, n, giving the number of atoms in the structure.

Line 2: A string reporting the serial number and the RMSD of the hit. When the file comes from a LINKER run, the RMSD refers to the metal-metal distance. When the file comes from an OVERLAY run, the RMSD refers to the superposition of the bonding vectors.

Lines 3 to 2+n: Each line describes an atom in the structure and contains the following information: the atom label and the x, y, and z coordinates.

4.0 HOW TO CITE HOSTDESIGNER IN THE LITERATURE

In publishing results obtained either in part of in full from use of HostDesigner, the user should use the following acknowledgment or citation:

Acknowledgment:

HostDesigner, Version 1.0 was developed by B. P. Hay and T. K. Firman in the William R. Wiley Environmental Molecular Sciences Laboratory at the Pacific Northwest National Laboratory under partial sponsorship of Chemical Sciences, Office of Basic Energy Sciences, Office of Science, U.S. Department of Energy under Contract No. DE-AC06-76RL01830. The software can be obtained at no charge from the following website: http://hostdesigner.emsl.pnl.gov.

Citation:

Hay, B.P.; Firman, T.K. "HostDesigner: a program for the *de novo* structure-based design of molecular receptors with binding sites that complement metal ion guests." *Inorg. Chem.* submitted.