Automation of conformational analysis and other molecular modeling calculations

Robin Taylor, Graham W. Mullier and Graham J. Sexton

ICI Agrochemicals, Jealott's Hill Research Station, Bracknell, Berkshire, UK

A software system has been developed for facilitating modeling calculations on large numbers of molecules. Using the system, it is possible to subject one or more molecules to a series of calculations, each requiring use of a different computer program. No user intervention is required: where necessary, output from one program is used automatically as input to the next. Names are assigned to output files automatically and in a systematic manner. As an example, the system can be used to perform a succession of calculations aimed at identifying the major low-energy conformers of each of a set of molecules, starting only from their chemical connectivities. The reliability of the results has been tested by calculations on 40 molecules taken from the Cambridge Structural Database. The observed crystal structure geometry could be found for the majority of these molecules.

Keywords: conformational analysis, Cambridge Structural Database, high-level modeling language

INTRODUCTION

In the last few years, molecular modeling has become an established method for facilitating the invention of drugs and agrochemicals.[1,2] A wide variety of modeling calculations is used, ranging from the simple (e.g., molecular mechanics energy minimization) to the more esoteric or computer-intensive (e.g., *ab initio* molecular orbital techniques, Comfa[3]). Typically, a molecular modeling group will perform many of these calculations with a single, general-purpose modeling package, such as SYBYL[4] or QUANTA.[5] However, there will always be a need to use several other, stand-alone programs.

Often, the output from one program must serve as input to another: for example, a molecule might be energy minimized by molecular mechanics and then subjected to a single-point *ab initio* molecular orbital calculation. There is a consequent need to develop interfaces between programs. In addition, it is sometimes necessary to perform the same sequence of calculations on a large number of molecules, e.g., when preparing data for a QSAR study.[6] Major benefits in efficiency can be obtained if a software environment is created that allows these operations to be performed as simply as possible.

We describe herein a system developed for this purpose. It is called SOFTI (SOFTware Integration), and is aimed at achieving the following objectives:

(1) Facilitate calculations on large numbers of molecules
(2) Provide a flexible mechanism for performing series of calculations sequentially
(3) Provide a uniform front end to programs, so that they may all be accessed in the same way
(4) Provide a log file mechanism, so that the user has a record of the calculations that have been performed and their outcomes
(5) Provide automatic mechanisms for naming files, so that the user can keep track of program input and output easily.

The first part of this paper describes the main features of SOFTI. This is followed by an example of its use in conformational analysis. Specifically, we show how it can be used to automate the following sequence of calculations on a large number of molecules:

(1) Initial generation of approximate three-dimensional structures
(2) Identification of chemical bonds about which torsional rotations can take place
(3) Exhaustive conformational searching, by scanning the bonds identified in the preceding step
(4) Selection of representative low-energy conformations from the output produced by the exhaustive search
(5) Final molecular-mechanics optimisation of these conformations.

Some sample results are given for a set of organic molecules taken from the Cambridge Structural Database.[7] It is shown that this sequence of calculations not only is easy to perform (i.e., can be performed without user intervention), but usually is capable of finding observed crystal structure conformations.

Address reprint requests to Dr. Taylor at ICI Agrochemicals, Jealott's Hill Research Station, Bracknell, Berkshire, RG12 6EY, UK.

Received 16 October 1991; accepted 19 November 1991

METHODS

SOFTI

Overview SOFTI consists of a number of routines, mainly for file handling, which are implemented under Unix[8] on a Silicon Graphics 4D/220 computer. The routines are largely written in C-Shell[9] and Awk.[10] Collectively, they act as a front end to a number of "procedures." Each procedure performs a particular type of calculation (e.g., Table 1). They vary in complexity. Thus, a simple procedure might process its input data with a single FORTRAN program. More complicated procedures may consist of two or more programs that run sequentially, the second operating on the results produced by the first, and so on.

SOFTI typically will be used in the following way. First, the user will select a set of molecules on which calculations are to be performed (a "workset"). This can be done within SOFTI, using a routine provided for the purpose. The user will then choose a procedure; for example, he might select AESOP (Table 1), a procedure for performing molecular mechanics structure optimization. Depending on circumstances, he may need to type in some instructions, e.g., to define convergence criteria for the energy minimization. Again, all of this can be done within SOFTI. Optionally, the user can select another procedure, which will operate on the results produced by the first. For example, having energy minimized the molecules, he may wish to calculate their electrostatic potentials. A third procedure then may be chosen, and so on, until the desired sequence of calculations is completely defined.

SOFTI then will be used to perform the specified calculations. The relevant SOFTI routine will select each molecule in turn, work out the names of the input files required by the first procedure, and the names of the output files that it will write, call the procedure with these file names as arguments, and write a suitable message to a log file when the procedure has run. Once all molecules have been subjected to the first procedure, a similar set of operations will be performed for the second and subsequent procedures. No user intervention is required.

Table 1. Some example procedures

Procedure	Purpose
AESOP	Performs molecular mechanics energy minimization (B.B. Masek, unpublished work)
CONCMOL	Uses CONCORD[11] to convert a 2D to a 3D molecular structure
CONPICKS	Analyzes results of an exhaustive conformational search
DOTS	Fills a protein cavity with a grid of solvent molecules[12]
GAMESS	Uses GAMESS[13] to perform an *ab initio* MO job
GAMESSD	Performs a direct SCF *ab initio* MO job
SIMIL	Calculates electrostatic similarity of pair of molecules[14]
SYBGENX	Creates and executes a SYBYL[4] command file
TORPICK	Identifies bonds about which torsional rotations can occur
ZMAT_SP	Creates a Z-matrix, used as input to GAMESS

File Naming Strict file-naming conventions are used in SOFTI. Each file name is considered to consist of four components, *viz.*,

directory/identifier_historyfield.extension

The first component is any valid Unix directory specification. The second component is a character string chosen by the user to identify the molecule, or other entity, to which the information contained in the file pertains, e.g., *mymolecule*. The fourth component is the file extension. In SOFTI, the user has no choice about which extension to use for a given file: it is uniquely and completely determined by the file format. For example, a file containing details of a molecule (atomic coordinates, etc.) in SYBYL "mol2" format[4] must have the extension *mol2*.

The remaining component of the file name is the history field. This contains information about how the file has been generated. Suppose, for example, that a molecule is built and stored in the file mymolecule.mol2 (where the directory has been omitted here and below for brevity), and then energy minimized with the molecular mechanics procedure AESOP. The optimized geometry could be written back to the file mymolecule.mol2, but this would overwrite the original geometry. The user may prefer therefore to write the optimized molecule to a new file. In SOFTI, this is done by requesting that a history field be added to the file name. The user is able to choose the name of the history field—in the above example, a suitable choice might be *aesop*. The optimized molecule thus would be written to a file called mymolecule_aesop.mol2.

History fields can be concatenated. For example, having energy minimized a molecule with AESOP, the user may wish to compute atomic partial charges with a procedure called CHARGE. The results could be written to yet another .mol2 file, with a suitable history field added, e.g., mymolecule_aesop_chrg.mol2.

Although history fields are generally under the user's control, there are some circumstances in which they will be added automatically. This is best illustrated by an example. Suppose that a molecule is built and saved in the file mymolecule.mol2, and then submitted to a procedure that performs a conformational search. This procedure identifies the low-energy conformers of the molecule and writes them out as separate .mol2 files. Clearly, the output files cannot all be called mymolecule.mol2. Nor is it known in advance how many output files there will be, since this depends on the results of the conformational search. SOFTI deals with this by adding a numerical history field to each output .mol2 file; this field varies from 1 for the first file to *n* for the *n*th file, i.e, mymolecule_1.mol2, mymolecule_2.mol2, etc.

Worksets In SOFTI, a workset is a list of objects that are to be subjected to the same calculation (i.e., operated on by the same procedure). Although the objects usually will be molecules, this is not always the case. For example, the procedure CONPICKS (Table 1) analyzes the listing of low-energy conformations produced by a conformational search

program. This listing is therefore the object operated on by CONPICKS.

Normally, each object will be stored in a separate file, e.g., moleculeA.mol2, moleculeB.mol2, etc. However, it is possible to have a workset that consists of molecules stored in a single SYBYL database.[4] In either case, the list of objects is stored in a simple text file called a workset file (extension *wset*), which contains the following information: the directory (or the full path name of the SYBYL database) in which the objects are stored; any number of lines of comments; the names of the files containing the objects, without directory or extension (or the names of the molecules in the SYBYL database). SOFTI contains a routine (CWSET), which allows the user to create worksets quickly and easily by use of wildcard characters.

Instruction Files Procedures obviously vary in the number and type of input files that they require. However, many need to read a file containing user-specified instructions about how the input data are to be manipulated. For example, a molecular mechanics procedure might well need an instruction file specifying what energy minimization algorithm is to be used, etc. This file is accorded a special status in SOFTI, and is called the instruction file. In a given job, a procedure that is called by SOFTI can read only one instruction file, though this file may contain separate instructions for each member of the workset (see below). Some procedures do not require an instruction file at all (i.e., there are no user-definable options). Others have default instruction files, which are used if the user does not specify an alternative.

There are two basic classes of instruction file: single and multiple. A multiple instruction file contains a separate block of instructions for each member of the workset; these instruction blocks are stored one after the other in the file. Alternatively, the same set of instructions may be required for each member of the workset. This might be the case, for example, if a number of molecules are to be energy minimized by a molecular mechanics program, and the same program options (convergence criteria, etc.) are required throughout. In this situation, the instruction file can be *single*, i.e., can contain just one block of instructions, to be used for all workset members.

Communication between SOFTI and Individual Procedures It is possible to explain how SOFTI determines the names of the input files required by a particular procedure, and the output files that it will produce. Suppose that a workset has been created containing molecules in the directory /usr/myarea, and that the workset is to be operated on by the procedure AESOP. Suppose, further, that the first member of the workset is molA. SOFTI contains a data file in which is stored the extensions of all the files read and written by each procedure. In the case of AESOP, they are:

- .aspins (instruction file)
- .mol2 (input file, contains a molecule whose geometry is to be optimized)
- .mol2 (output file, contains optimized molecule)
- .asp_log (line printer file, containing details of the optimization).

It is straightforward for SOFTI to work out that the input mol2 file for the first workset member must be called /usr/myarea/molA.mol2. The same instruction file will be used for all members of the workset, though it may contain a separate block of instructions for each member (see above). The name of the instruction file will have been specified by the user when the SOFTI job was created. Also at this point, the user will have been asked whether history fields are to be added to any output files. Suppose that the history field *aes* was specified for output files with the extension *mol2*, but no history field was specified for any other type of output file. The names of the files produced by AESOP for the first workset member are evidently /usr/myarea/molA_aes.mol2 and /usr/myarea/molA.asp_log.

Jobsets and the Automatic Creation of New Worksets A jobset is a text file (extension *jset*), which defines a series of calculations (jobs). Each job involves: a workset, defining the objects that are to be subjected to the calculation; a procedure name, defining the type of calculation that is to be performed; an instruction file, giving additional information to the procedure, e.g., about program options that are to be used. The jobset also contains information about any history fields that are to be added to the output file names. SOFTI contains a routine (CJSET) that allows the user to create jobsets quickly and easily by means of a question-and-answer dialogue.

A simple jobset file is listed in Table 2. It contains a

Table 2. Example jobset

Line	Contents of jobset file		Remarks
1			
2	/usr/mydirectory/example.wset		Workset to be operated on
3	AESOP		Procedure to be used
4	/usr/mydirectory/my.aspins		Instruction file to be used
5	#_aesop.mol2	#.asp_log	Output files to be produced
6	example_2.wset	none	New worksets to be created
7			
8	/usr/mydirectory/example_2.wset		Workset to be operated on
9	CHARGE		Procedure to be used
10	none		No instruction file needed
11	#.mol2	#.charge_log	Output files to be produced
12	none	none	No new worksets to be created

series of jobs (two in this case), each separated from the others by a blank line (lines 1, 7). Each job defines a particular calculation, to be performed on a particular workset. Thus, the first job will use the procedure AESOP (line 3) to perform molecular mechanics energy minimization on every molecule in the workset example.wset (line 2; directory omitted here and subsequently for brevity). This workset must have been created already by the user. If it contains, say, molA and molB, the user must have previously built these two molecules and saved them in the files molA.mol2 and molB.mol2. An instruction file, my.aspins, will be used to specify various AESOP program options (line 4). Two output files will be produced for each workset member (line 5; # is a generic symbol for a workset member). These will have the extensions mol2 and asp_log. The output .mol2 file will contain the energy-minimized structure and the .asp_log file will contain details of the optimization. The user has specified that the history field *aesop* will be added to the former (line 5). Thus, for example, AESOP will produce the output files molA_aesop.mol2 and molA.asp_log by processing the input file molA.mol2.

The next job will use a procedure CHARGE (line 9) to compute atomic partial charges. However, the user wishes this calculation to be performed on the optimized molecules. It is therefore necessary to create a new workset containing the members molA_aesop and molB_aesop. The last line of the AESOP jobset entry (line 6) instructs SOFTI to create this workset automatically and store it in example_2.wset. This workset will contain the names of all output files produced by AESOP with the extension *mol2*. No new workset is created from the .asp_log output files; there would be no point, because it would be identical to the original workset, example.wset (since no history field is being added to the .asp_log files).

The second job therefore reads the .mol2 files produced by AESOP (line 8), *viz.*, molA_aesop.mol2 and molB_aesop.mol2. CHARGE requires no instruction file; hence, *none* is specified on line 10. Output files with the extensions *mol2* and *charge_log* will be produced, no history field being added (line 11). Thus, for example, CHARGE will produce the output files molA_aesop.mol and molA_aesop.charge_log by processing the input file molA_aesop.mol2. The new .mol2 file will overwrite the input file and will contain the AESOP-optimized molecule, with partial atomic charges. The charge_log file contains details of the charge calculation. No new worksets are created by this job (line 12).

The calculations specified in a jobset are performed by the SOFTI routine EJSET. EJSET will process each workset member in turn with the first procedure, generating file names and new worksets as required. Once this has been done, the second job will be run, and so on. The outcome of each procedure call is written to a log file, so that the user has a complete record of the set of calculations that has been performed.

Use of SYBYL within SOFTI The principal modeling package in use at ICI Agrochemicals is SYBYL.[4] Since this program has a wide range of functionality, it is important that it can be used from within SOFTI. This is achieved by the procedure SYBGENX (Table 1), which is used to create and execute a SYBYL command file that will perform a series of calculations on each molecule in a workset. For example, suppose that it is desired to read each molecule into SYBYL, minimize its energy, calculate partial charges using the Gasteiger method, calculate the electrostatic potential, and then save the results. Using SYBGENX, this is done by creating a file containing the necessary SYBYL commands for performing this sequence of operations on one molecule. This file is called a *SYBYL pseudo-sequence* and has the extension *psyb*. In SOFTI terms, it acts as the instruction file (see above) for the procedure SYBGENX. SYBGENX takes the .psyb and workset files, and creates a new file containing all of the SYBYL commands required to perform the desired sequence of calculations on every molecule in the workset. SYBGENX then calls SYBYL to execute these commands.

A job using SYBGENX can be included in a SOFTI jobset containing jobs that use other procedures. Thus, input to the SYBGENX job can be preprocessed by other procedures accessible via SOFTI, and output can be postprocessed. The value of SYBGENX is further increased by the fact that SPL (SYBYL Programming Language[4]), C-Shell, and other routines can be included in a .psyb file.

Other Features The basic features of SOFTI have been described. They are fully implemented and have been used successfully for several months. This section describes a number of additional features, some of which are still under development.

Worksets can be used in various ways. The simplest way is one_at_a_time. This means that each workset member is selected in turn by SOFTI and passed to the procedure for processing. This would be the case for a procedure such as AESOP. However, it is possible to envisage several other ways of using worksets, some of which are already implemented in SOFTI, and some of which may be added later. For example, workset members may be passed two_at_a_time to a procedure. This would be the case if the procedure calculated a similarity index for a pair of molecules and we wished to compare the first workset member with the second, the third with the fourth, and so on.

A SYBYL pseudo-sequence (see above) that performs a popular combination of calculations is likely to be of general value. The same is true of instruction files that specify common choices of program parameters for programs offering complex options. For this reason, instruction files (including SYBYL pseudo-sequences) can be saved and documented for general use. Jobsets can also be saved for general use: an example of one such standard jobset is described next.

Automated Conformational Analysis

This section describes a series of calculations that can be performed without user intervention by creating and executing a single SOFTI jobset. The aim of the calculations is to identify the most important low-energy conformers of each of a set of of n molecules, starting from the chemical connectivities of the molecules. The overall strategy is summarized above. Each job in the jobset is described in more detail below.

Initial Generation of Approximate 3-D Structures Each molecule is initially defined by its SMILES string, which is a linear sequence of characters completely specifying chemical connectivity.[15] The SMILES strings are stored,

one per line, in a single SMILES file (extension *smi*), which is submitted to the procedure CONCMOL. This procedure calls the program CONCORD, which generates a preliminary three-dimensional structure for each of the *n* molecules, using a rule-based methodology.[11] The resulting structures are written out as separate files (one per molecule), in SYBYL mol2 format.

Identification of Rotatable Bonds In the next job, each .mol2 file is submitted to the procedure TORPICK. This runs an in-house program designed to identify the chemical bonds that should be scanned in an exhaustive conformational search. Thus, the program begins by analyzing the molecular connectivity to identify single, acyclic bonds. Of these, bonds to terminal atoms or linear terminal groups (e.g., —C≡CH) are removed. For each remaining bond, the symmetries of the groups at either end of the bond are determined, to deduce the angular range through which the bond should be scanned. For example, a bond between two asymmetric groups must be rotated through 360°, whereas a bond to a trifluoromethyl group need only be scanned through 120°.

The scan step size is then set to 30° for each rotatable bond. The total number of conformations that would be generated in the search is calculated. If this is too large (greater than 10^7 in our implementation), it is reduced to an acceptable value by successively applying the following steps:

(1) Constraining ester linkages, un-ionized carboxylic acids and unsubstituted and monosubstituted amides to be planar
(2) Disabling the rotation of methyl and $—NH_3^+$ groups
(3) Increasing the scan step size for each bond from 30° to 45° and then to 60°
(4) Disabling the remaining torsion axes one by one, starting from the least important (i.e., those at the periphery of the molecule).

Finally, dummy atoms are added to enable rotation around bonds adjacent to acetylenic linkages.

The output produced by TORPICK is a new .mol2 file, identical to the input file except that it contains the information pertaining to the rotatable bonds.

Exhaustive Conformational Searching In the next job, the procedure SYBGENX is used with the standard instruction file SEARCHSYB.psyb; this causes an exhaustive conformational search[16] to be performed on each of the *n* molecules. These searches are performed using the SYBYL SEARCH option[4] and the information about rotatable bonds deduced in the preceding job. For each molecule, the output consists of a listing of the torsion angles of all conformations estimated to be within a certain energy (we use 3 kcal/mole) of the global minimum. Since the energy calculations at this stage are necessarily crude (because of the large number of conformations that must be examined), the results are only approximate and must be processed further (see below).

Unfortunately, the SYBYL SEARCH program is difficult to use in a routine, automatic manner. Various program parameters must be set (e.g., the reduction factors to be applied to van der Waals radii in the initial prescreening of conformations for steric clashes[4]), and the optimum values of these parameters tend to vary from one molecule to another. A compromise set of parameters has been defined in SEARCHSYB.psyb (*viz.*, general van der Waals reduction factor = 0.9; conformation limit = 50000; maximum energy difference = 3.0; no electrostatics; reference conformation = current[4]). However, these settings are not suitable for all molecules. Occasionally, therefore, the program fails to locate low-energy conformations. Work is in hand to develop software that will rectify this problem; in the meantime, conformational searching is the weakest link in this series of calculations.

Selection of Representative Low-Energy Conformations for Further Optimization The need for further processing of the SYBYL SEARCH output is best illustrated by a simple, hypothetical example. Consider a molecule with just one rotatable bond, the conformation about which is measured by the torsion angle τ; assume that the energy of the molecule varies with τ as shown in Figure 1. Now suppose that SYBYL SEARCH is used to perform an exhaustive conformational search, scanning τ from $-180°$ to $+180°$ in 45° steps. The program will estimate (rather crudely) the

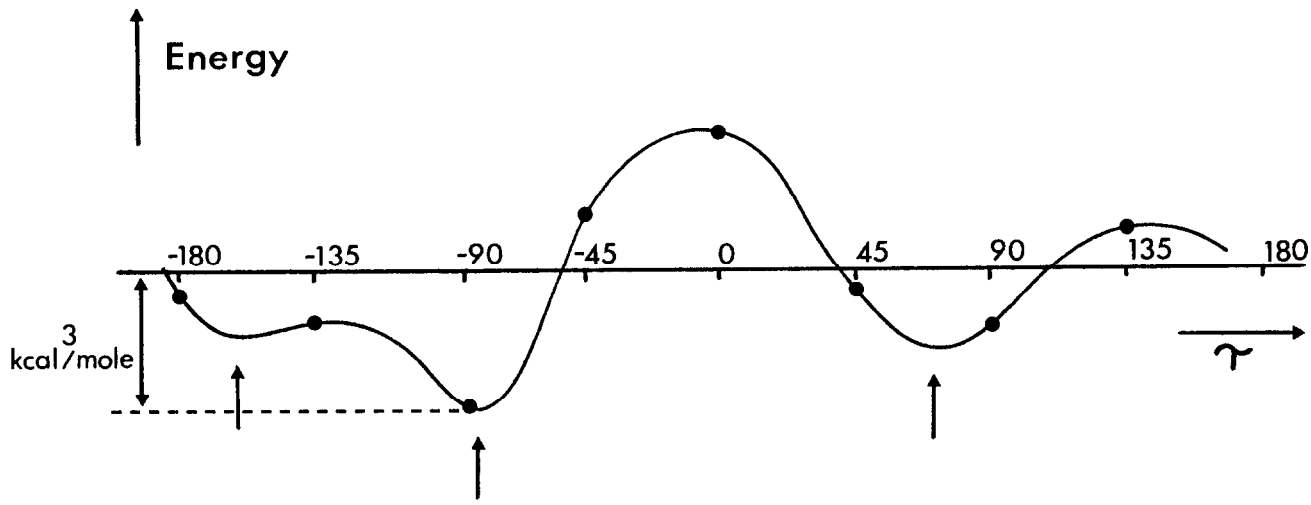

Figure 1. Example potential energy profile

energies at $\tau = -180, -135, -90°$, etc., and will list all conformations that fall within 3 kcal/mole (in our implementation) of the lowest energy geometry thus found. These conformations will be $-180, -135, -90, +45$ and $+90°$.

The SYBYL SEARCH output therefore identifies low-energy regions of conformational space, but in a rather inefficient manner. The information that is really required is the locations of the various potential energy minima, i.e., the points arrowed in Figure 1. These can be identified by molecular mechanics optimization of the conformations listed out by SYBYL SEARCH. However, optimization of *all* of these conformations is inefficient, since some will converge to the same minimum (e.g., $\tau = -180$ and $-135°$). It is therefore desirable to select only a subset for minimization—ideally, one conformation from each potential energy well. Subset selection is essential in most real situations, when the potential energy space is usually multidimensional (i.e., several torsion angles are scanned) and the SYBYL SEARCH output may consist of tens or hundreds of thousands of conformations.

Selection of a set of representative conformations from each SYBYL SEARCH output file is achieved with the procedure CONPICKS. This calls an in-house program which reads in the SEARCH output file, together with the corresponding .mol2 file. The program performs the following operations:

(1) The conformations listed in the SEARCH output are grouped into "families." Each family consists of a group of related conformations: formally, each conformation in a family must be contiguous to at least one other member of the family, and to no member of any other family. Contiguous conformations are next to each other in the grid used for the search (e.g., in Figure 1, -180 is contiguous to -135 but not $-90°$). The SEARCH output generated from the hypothetical example discussed above (Figure 1) would be divided into two families, one containing the conformations $-180, -135$ and $-90°$, and the other containing $+45$ and $+90°$.

In testing for contiguity, the program takes into account the periodicity of torsion angles (e.g., $\tau = -180$, being identical to $\tau = +180$, is considered contiguous to both -135 and $+135°$). The range through which a bond was scanned is assumed to define its symmetry. Thus, a torsion angle that was scanned through only 180° is assumed to be two-fold symmetric, so that torsions of 0° and 180° are identical, and both contiguous to, say, 135° (assuming the step size is 45°).

In multidimensional space, there are several ways of defining contiguity. For example, in a three-axis scan with 30° steps, the conformation (30, 60, 90) would always be considered contiguous to (60, 60, 90) and (30, 90, 90). However, it might also be considered contiguous to (60, 90, 90) or even (60, 90, 120). This is geometrically equivalent to cubes being considered contiguous if their faces touch, their edges touch, or if their corners touch. In space of any dimensionality, the user is allowed to control how contiguity is defined. By default, the least stringent criterion (equivalent to "corners touching") is used.

(2) Details are printed of the most populous families and those containing the lowest energy conformations. Thus, information is given about the lowest and highest energy conformations in each family, the number of conformations and discrete potential energy wells that the family appears to contain, and the distribution of torsion angles within the family. These data provide a useful summary of the conformational space of the molecule.

(3) A set of conformations is chosen for further molecular mechanics optimization. The user is allowed to specify the minimum and maximum number of conformations that can be selected. Within these constraints, the program attempts to choose conformations from as many families as possible and, within a family, from as many different potential energy wells as possible. For example, if instructed to select three conformations from the hypothetical SYBYL SEARCH output discussed above, the program would select $-135, -90$ and $+90°$.

If the user specifies that no more than m conformations are to be chosen, but more than m potential energy wells appear to exist, the program will select as diverse a set of conformations as possible. In estimating the dissimilarity of each pair of conformations, more weight is given to differences in "important" torsion angles (e.g., those at the center of the molecule) than to unimportant torsions (e.g., those specifying the rotational orientation of methyl groups).

(4) The chosen set of conformations is written out as a single .mol2 file. This file is referred to here as a "concatenated" .mol2 file, since it contains several different geometries of the same molecule.

Final Molecular Mechanics Optimization The final step is straightforward. Each of the n concatenated .mol2 files is submitted to the procedure AESOPM. This calls the molecular mechanics program AESOP to energy minimize each of the conformations selected in the preceding job. The resulting optimized geometries are sorted into ascending order of energy, and duplicates (should two starting conformations converge to the same minimum) are eliminated. The final set of sorted, optimized conformers are written out to a new concatenated .mol2 file, which may subsequently be read into SYBYL for visual inspection.

RESULTS AND DISCUSSION

Using SOFTI, the series of calculations described above can be performed without user intervention, even for a large number of molecules. However, it is obviously necessary for it to produce reliable results. The accuracy of conformational predictions can best be tested with reference to experimental data. The SOFTI jobset described above was therefore used to predict the low-energy conformers of 40 molecules whose crystal structures have been determined by X-ray diffraction. While the observed crystal structure geometry of a molecule is not necessarily its global minimum conformation, it is obviously energetically accessible and should be recognizably similar to one of the conformers identified by the theoretical calculations.

The test set was taken from the Cambridge Structural Database.[7] Molecules were not selected entirely at random, since it was desired to have a variety of molecular flexi-

Table 3. Molecules used for conformational analysis calculations

Cambridge Refcode[7]	Ref.	Compound name
ANSFON	(a)	Dianiline-sulphone
BAFPED	(b)	1-Bromoacetyl-4-chlorobenzene
BAKRAG	(c)	2-(2′-Pyridylthio)-3-nitropyridine
BANHAZ	(d)	7-(p-N,N-Dimethylaminophenyl)-7,8,8-tricyanoquinodimethane
BEBLID	(e)	N-(p-Chlorophenyl)-N′-methoxy-N′-methyl-urea
BEDLUR	(f)	Ethyl-N-(2-amino-6-(4-fluorophenylmethylamino)-pyridin-3-yl) carbamate
BEFTEL	(g)	Phenoxyacetic acid
BEFTOV01	(h)	2-(4-Chlorophenoxy)-propionic acid
BENTUJ	(i)	4′-Cyanophenyl-4-n-pentoxybenzoate
BEVPEX	(j)	4-Nitro-2,6-diphenylphenol
BIGGIH	(k)	p-Tricyanovinyl-N-ethyl-N-(beta-cyanoethyl)-aniline
BIGXAQ	(l)	1,3-Diphenylpropane-1,2,3-trione
BILHAF	(m)	1-(4-Fluorophenyl)-4-(4-hydroxy-4-(4-methylphenyl)-1-piperidinyl)-butan-1-one
BILSEU	(n)	N-Methoxycarbonyl sulphanilamide
BINMAM	(o)	N-(o-Bromophenyl)-o-bromobenzylamine
BINROF	(p)	2,4-Dinitrobenzaldehyde-(1H-1-tetrazol-5-yl) hydrazone
BIWWEJ	(q)	5-(2-Benzylaminoethyl)-3-phenyl-pyrazole
BOCNAI	(r)	E-1-(2-Bromo-4,5-dimethoxyphenyl)-2-(3,4-dimethoxyphenyl)-ethylene
BOCRIU	(s)	5,5-Dimethylcyclohexane-1,2,3-trione-2-(4-methylphenylhydrazone)
BODCEC	(t)	2,5-Hexanediyl dibenzoate
BOLFIR	(u)	2-(2,6-Dichlorophenylcarbamoyl)-benzoic acid
BOLFOX	(v)	2-(4-(4-Chlorophenoxymethyl)-phenoxy)-propionic acid
BOMKOD	(w)	2-Bromo-2-nitro-1,3-propanediol
BOMKUJ	(x)	1,1,1-tris(Hydroxymethyl)propane
BOWHIE	(y)	N-p-Toluoyl-N-methyl-hydroxylamine
BOWNEG	(z)	(4-Chlorophenoxyacetyl)-4-methylbenzene
BUGPE	(aa)	2,3-Butanediyl dibenzoate
BZCPTI11	(bb)	2-(p-Methylbenzyl)-5-(p-bromobenzylidene)-cyclopentanone
FPAMCA11	(cc)	2-((3-(Trifluoromethyl)phenyl)amino)-benzoic acid
HETPAL01	(dd)	bis(2-Hydroxyethyl)-terephthalate
MBZYAN03	(ee)	p-Methyl-N-(p-methylbenzylidene)-aniline
MDECBZ10	(ff)	4a,8a-Dimethyl-trans-decal-1-yl p-bromobenzoate
NOPHTE10	(gg)	2,4-Dinitrodiphenylsulphide
OVERAT01	(hh)	2,3-Dimethoxybenzoic acid
PYMSBZ11	(ii)	N-((1-Ethyl-2-pyrrolidinyl)-methyl)-2-methoxy-5-sulphamoyl-benzamide
SLFNMA01	(jj)	4-Methyl-2-sulphanilamido-pyrimidine
THXMAM01	(kk)	tris(Hydroxymethyl)-aminomethane
TOLAMA10	(ll)	tris(p-Tolyl)-amine
ZZZPUS02	(mm)	1-n-Butyl-3-p-toluenesulphonylurea
ZZZPZE01	(nn)	3,3′-Thiodipropionic acid

(a) L.G. Kuz'mina, Yu.T. Struchkov, N.V. Novozhilova, G.L. Tudorovskaya, *Kristallografiya* **26**, 695, 1981; (b) K.N. Prasad, *Cryst. Struct. Commun.* **10**, 879, 1981; (c) M. Kimura, S.H. Simonsen, S.R. Caldwell, G.E. Martin, *J. Heterocycl. Chem.* **18**, 469, 1981; (d) E.G. Popova, L.A. Chetkina, B.P. Bespalov, *Zh. Strukt. Khim.* **22**, 132-4, 1981; (e) Ya.M. Nesterova, M.A. Porai-Koshits, A.G. Moev, *Zh. Strukt. Khim.* **22**, 111-5, 1981; (f) W. von Bebenburg, K. Thiele, J. Engel, W.S. Sheldrick, *Chem. Zeit.* **105**, 217, 1981; (g) C.H.L. Kennard, G. Smith, A.H. White, *Acta Cryst.* **B38**, 868, 1982; (h) S. Raghunathan, K. Chandrasekhar, V. Pattabhi, *Acta Cryst.* **B38**, 2536, 1982; (i) U. Baumeister, H. Hartung, M. Gdaniec, M. Jaskolski, *Mol. Cryst. Liq. Cryst.* **69**, 119, 1981; (j) S. Ueji, K. Nakatsu, H. Yoshioka, K. Kinoshita, *Tetrahedron Lett.* **23**, 1173, 1982; (k) Z.P. Povet'eva, L.A. Chetkina, B.P. Bespalov, *Zh. Strukt. Khim.* **23**, 168-2, 1982; (l) R.L. Beddoes, J.R. Cannon, M. Heller, O.S. Mills, V.A. Patrick, M.B. Rubin, A.H. White, *Aust. J. Chem.* **35**, 543, 1982; (m) B. Tinant, G. Germain, J.P. Declercq, M. van Meerssche, M. Azibi, M. Draguet-Brughmans, R. Bouche, *Bull. Soc. Chim. Belg.* **91**, 283, 1982; (n) Lu Guangying, Yang Qingchuan, Zhang Zeying, Yang Huazheng, *Acta Chim. Sinica* **40**, 476, 1982; (o) A.F. Berndt, E.O. Schlemper, *Acta Cryst.* **B38**, 2493, 1982; (p) D.S.S. Gowda, R. Rudman, K.R. Acharya, *Acta Cryst.* **B38**, 2487, 1982; (q) R. Kunstmann, E.F. Paulus, *Angew. Chem. Int. Ed. Engl.* **21**, 548, 1982; (r) J.M. Arrieta, E. Lete, E. Dominguez, G. Germain, J.P. Declercq, J.M. Amigo, *Acta Cryst.* **B38**, 3155, 1982; (s) M.G.B. Drew, B. Vickery, G.R. Willey, *J. Chem. Soc. Perkin 2*, **1982**, 1297; (t) G. Bocelli, M.F. Grenier-Loustalot, *Acta Cryst.* **B38**, 3135, 1982; (u) C.H.L. Kennard, G. Smith, G.F. Katekar, *Aust. J. Chem.* **35**, 1933, 1982; (v) G. Smith, C.H.L. Kennard, W.L. Duax, D.C. Swenson, *Aust. J. Chem.* **35**, 2151, 1982; (w) D.S.S. Gowda, R. Rudman, *J. Chem. Phys.* **77**, 4666, 1982; (x) D.S.S. Gowda, N. Federlein, R. Rudman, *J. Chem. Phys.* **77**, 4659, 1982; (y) V.N. Kalinin, M.Yu. Antipin, V.M. Yurchenko, Yu.T. Struchkov, *Zh. Strukt. Khim.* **23**, 83-5, 1982; (z) T. Atabaev, Yu.V. Gatilov, N.V. Podberezskaya, A. Ashirov, S.V. Borisov, *Zh. Strukt. Khim.* **23**, 179-5, 1982; (aa) G. Bocelli, M.F. Grenier-Loustalot, *Acta Cryst.* **C39**, 633, 1983; (bb) C.R. Theocharis, W. Jones, M. Motevalli, M.B. Hursthouse, *J. Cryst. Spectrosc.* **12**, 377, 1982; (cc) H.M. Krishna Murthy, T.N. Bhat, M. Vijayan, *Acta Cryst.* **B38**, 315, 1982; (dd) W.S. McDonald, E.L.V. Lewis, D.I. Bower, *Acta Cryst.* **C39**, 410, 1983; (ee) I. Bar, J. Bernstein, *Acta Cryst.* **B38**, 121, 1982; (ff) R.E. Ireland, M.I. Dawson, C.J. Kowalski, C.A. Lipinski, D.R. Marshall, J.W. Tilley, J. Bordner, B.L. Trus, *J. Org. Chem.* **40**, 973, 1975; (gg) V. Cody, P.A. Lehmann, *Cryst. Struct. Comm.* **11**, 1671, 1982; (hh) R.F. Bryan, D.H. White, *Acta Cryst.* **B38**, 1012, 1982; (ii) L.Y.Y. Ma, N. Camerman, A. Camerman, *Acta Cryst.* **B38**, 2861, 1982; (jj) K.R. Acharya, K.N. Kuchela, G. Kartha, *J. Cryst. Spectrosc.* **12**, 369, 1982; (kk) E. Kendi, *Z. Krist.* **160**, 139, 1982; (ll) S.L. Reynolds, R.P. Scaringe, *Cryst. Struct. Comm.* **11**, 1129, 1982; (mm) J.D. Donaldson, J.R. Leary, S.D. Ross, M.J.K. Thomas, C.H. Smith, *Acta Cryst.* **B37**, 2245, 1981; (nn) K. Prout, S. Hernandez-Cassou, *Acta Cryst.* **B38**, 338, 1982

Table 4. RMS and maximum deviations between non-hydrogen atoms of observed and theoretical geometries

Cambridge Refcode	Deviation (Å) RMS	Max.	Energy†	Comments
ANSFON	0.32	0.58	0.0	Three similar molecules in asymmetric unit; molecule containing S1 used in fit
BAFPED	0.12	0.23	1.4	
BAKRAG	0.72	1.14	3.5	
BANHAZ	0.84	1.79	0.0	
BEBLID	0.34	0.57	0.0	
BEDLUR		NOT FOUND		
BEFTEL	0.08	0.15	0.0	
BEFTOV01	0.18	0.35	0.6	
BENTUJ		NOT FOUND		
BEVPEX	0.09	0.19	0.0	
BIGGIH		NOT FOUND		
BIGXAQ		NOT FOUND		
BILHAF	0.23	0.58	0.1	Two similar molecules in asymmetric unit; molecule with primed atom names used in fit
BILSEU	0.15	0.30	0.0	
BINMAM		NOT FOUND		
BINROF	0.49	1.23	0.9	
BIWWEJ	0.25	0.50	2.1	
BOCNAI	0.33	1.30	0.0	RMS devn = 0.14 if one—OMe group omitted
BOCRIU	0.20	0.40	0.0	
BODCEC	0.40	0.67	1.5	
BOLFIR	0.40	1.23	0.2	
BOLFOX		NOT FOUND		
BOMKOD	0.06	0.11	2.2	
BOMKUJ	0.08	0.17	1.9	
BOWHIE	0.07	0.11	1.1	
BOWNEG	0.12	0.18	0.6	
BUGPEY	0.35	0.71	0.1	
BZCPTI11	0.69	1.61	0.0	RMS devn = 0.13 if one aromatic ring omitted
FPAMCA11	0.53	1.26	0.2	RMS devn = 0.29 if trifluoromethyl Fs omitted
HETPAL01	0.24	0.53	1.1	
MBZYAN03	0.22	0.35	0.0	
MDECBZ10	0.26	0.51	0.0	
NOPHTE10	0.52	0.91	0.0	Two similar molecules in asymmetric unit; molecule without primed atom names used in fit
OVERAT01	0.14	0.29	0.8	
PYMSBZ11	0.56	1.61	2.4	RMS devn = 0.42 if one—OMe group omitted
SLFNMA01	0.56	1.39	0.0	
THXMAM01	0.08	0.12	2.1	
TOLAMA10	0.20	0.42	0.0	Two similar molecules in asymmetric unit; molecule containing N2 used in fit
ZZZPUS02	0.68	1.23	0.1	RMS devn = 0.52 if n-butyl chain omitted
ZZZPZE01		NOT FOUND		

†Energy = $E(m) - E(g)$, where $E(m)$ is the calculated energy of the theoretical conformer that best matches the observed crystal structure geometry, and $E(g)$ is the calculated energy of the most stable theoretical conformer found (in kcal/mole)

bilities in the test set. Thus, some comparatively rigid molecules were chosen (two or three variable torsions), together with a number of very flexible systems (ten or more torsions), and several of intermediate flexibility. However, molecules were not chosen because they were perceived to be particularly easy or difficult for the theoretical conformational predictions. Details of the test set are given in Table 3.

The low-energy conformers of the molecules were predicted as described above and compared with the observed geometries, using SYBYL. For each molecule, the predicted conformer that was most similar to the crystal structure geometry was thus identified. Results are summarized in Table 4. For each molecule, this gives the root-mean-square (rms) and maximum deviation between non-hydrogen atoms, when the observed geometry is overlaid on the most similar theoretical conformer. The calculated energy of this conformer, relative to the global minimum found, is also given.

The calculations failed to find the crystal structure conformation on seven occasions, presumably because the relevant potential energy minimum was not identified in the conformational search. However, a conformation closely related to that observed was found for several of these molecules. In another six cases (BAKRAG, BANHAZ, BZCPTI11, NOPHTE10, SLFNMA01, ZZZPUS02), the conformational minimum observed in the crystal structure was successfully located, but the rms deviation between theoretical and observed geometries exceeded 0.5 Å. It is difficult to say whether these discrepancies are due to inaccuracies in the molecular mechanics predictions or the effect of crystal-packing forces. However, the latter are implicated for at least one molecule, BANHAZ, which is more nearly planar in the crystalline state than is predicted theoretically. Stabilization of strained, planar conformations by systematic crystal-packing forces is known to occur in molecules such as biphenyl,[17] and a similar phenomenon may well be occurring here.

The geometries of the remaining 27 molecules were reproduced very accurately, except for trivial errors in BOCNAI and PYMSBZ11 (orientation of one methoxy substituent incorrect) and in FPAMCA11 (discrepancy in rotational orientation of trifluoromethyl group). Interestingly, the energy of the crystal structure conformer was calculated to be within 2.5 kcal/mole of the global minimum for all of these molecules, and within 1.0 kcal/mole for 18 of them. Overall, we regard these results as encouraging.

CONCLUSIONS

Until recently, software development in molecular modeling was focused primarily on the production of large, multifunctional graphics packages. Although of great value, these packages can never provide all of the functions required by molecular modelers. Consequently, there will always be a need for a variety of separate programs, each of which performs a specific task. For optimum use, it is desirable to integrate these programs with one another, and with the large multifunctional packages, so that a molecule can be subjected to a series of calculations automatically, even if this requires sequential execution of several different programs.

This integration has been achieved by the SOFTI system described above, as illustrated by the conformational analysis example. The major value of SOFTI is that it allows different programs to be linked together in an almost unlimited variety of ways. To this extent, SOFTI may be regarded as a high-level molecular modeling language.

ICI will be happy to consider requests for SOFTI from academic institutions.

ACKNOWLEDGEMENTS

We thank J.S. Delaney, K.J. Heritage, A. Mullaley, and R.C. Viner for helpful discussions. ICI Agrochemicals in the UK is part of Imperial Chemical Industries plc.

REFERENCES

1 Cohen, N.C., Blaney, J.M., Humblet, C., Gund, P. and Barry, D.C. *J. Med. Chem.* 1990, **33**, 883–894
2 Odell, B. *J. Comput.-Aid. Molec. Design,* 1988, **2**, 191–216
3 Cramer, R.D. III, Patterson, D.E. and Bunce, J.D. in *QSAR: Quantitative Structure-Activity Relationships in Drug Design* (J.L. Fauchere, ed.) Liss, New York, 1989, pp. 161–165
4 *SYBYL Users Manual, 1991* Tripos Associates, St. Louis, MO 63117, USA
5 Polygen Corporation, Waltham, MA 02254, USA
6 Glen, R.C. and Rose, V.S. *J. Mol. Graph.* 1987, **5**, 79–86
7 Allen, F.H., Bellard, S., Brice, M.D., Cartwright, B.A., Doubleday, A., Higgs, H., Hummelink, T., Hummelink-Peters, B.G., Kennard, O., Motherwell, W.D.S., Rodgers, J.R. and Watson, D.G. *Acta Cryst.* 1979, **B35**, 2331–2339
8 Coffin, S. *Unix: The Complete Reference* McGraw-Hill, Berkeley, 1988. Unix is a trademark of AT&T/Bell Laboratories
9 Anderson, G. and Anderson, P. *The Unix C-Shell Field Guide* Prentice-Hall, Englewood Cliffs, 1986
10 Aho, A.V., Kernighan, B.W. and Weinberger, P.J. *The AWK Programming Language* Addison–Wesley, Reading, 1988
11 Rusinko, A., III, Skell, J.M., Balducci, R., McGarity, C.M. and Pearlman, R.S. *CONCORD, a Program for the Rapid Generation of High Quality Approximate 3-Dimensional Molecular Structure* University of Texas at Austin and Tripos Associates, St. Louis, 1988
12 Delaney, J.S. *J. Mol. Graph.* 1992, **10**, 174–177
13 Guest, M.F. and Sherwood, P. *GAMESS User's Guide and Reference Manual* SERC Daresbury Laboratory, UK, 1990
14 Richards, W.G. and Hodgkin, E.E. *Chem. Brit.* 1988, **24**, 1141–1144
15 Weininger, D. and Weininger, J.L. in *Comprehensive Medicinal Chemistry* (C.A. Ramsden, ed.) Pergamon, Oxford, 1990, vol. 4, pp. 59–82
16 Motoc, I., Dammkoehler, R.A. and Marshall, G.R. in *Mathematical and Computational Concepts in Chemistry* (N. Trinajstic, ed.) Ellis Horwood, Chichester, 1986, pp. 222–251
17 Brock, C.P. and Minton, R.P. *J. Am. Chem. Soc.* 1989, **111**, 4586–4593