- TOM allows the use of any type of ligand molecule
- energy minimization is performed interactively
- all the FRODO options are present
- hydrogen atoms are not included

DESCRIPTION OF THE RESIDUE DICTIONARY

The residue description (amino-acids and special groups such as the coenzyme NADH) is partly derived from FRODO's dictionary. Some additional parameters which are used for energy calculation of fitting have been added. They are connectivities, partial charges and Lennard-Jones parameters. The ligand is described in the same way as in the GAUSS *Z*-matrix[3]. Partial atom charges, Lennard-Jones parameters and a special ring parameter to restrict certain dihedral angles have also been added. These data are kept in one file (*Z*-file) for each ligand molecule.

LIGAND COORDINATE FILES

A newly created ligand is automatically inserted in the protein at a chosen position when running TOM. It can then be moved and transformed using the display. Current coordinates are kept in the FRODO file DSN2. The *Z*-file and FRODO's DSN2 file communicate in the following way. The *Z*-file is updated from DSN2 when leaving the display mode. DSN2 is updated when coming into display mode if the *Z*-file has been modified by off display options.

LIGAND FITTING

Ligand fitting can be carried out using (apart from the standard FRODO options FBRT, TOR, MOVE) the BELL option displaying the shortest contacts for the moving atoms, as well as the ENER option which performs energy calculation. It can also be achieved through an automatic procedure FIT which minimizes the energy (Lennard-Jones, torsion and charge potentials) using a conjugate-gradient[4] procedure. The ligand and all the residues within an R1 radius from all the ligand atoms are flexible, all residues within an R2 radius from all the previously chosen moveable atoms form a fixed shell to prevent an 'explosion' of the flexible zone. The conformational space of the ligand and all the flexible residue side chains is explored by FIT. The torsional angles are the minimization parameters. In addition, translation/rotation of the ligand can be included if necessary. During the FIT procedure, a view of the working zone is displayed, and the calculation can be stopped at any time by a command before the convergence criterion is reached.

FIT can also be used for model building of proteins. The *Z*-file is not needed to run TOM, and any amino acid can be treated as the 'ligand' molecule so that a zone of the protein can be energy refined.

The speed of FIT makes it very attractive. For a molecule having 6 torsional angles within a 400 atoms fixed zone, convergence was reached after 20 steps of conjugate gradient, using about one minute of real time and 30 s CPU time.

FUTURE PROJECTS

We plan to speed up the FIT option even more and to include surfacing of both active site and ligand.

REFERENCES

1 **Jones, T A** in **Sayre, D (ed)** *Computational crystallography* Clarendon Press, Oxford (1982) pp 307–317
2 **Bussetta, B, Tickle, I J and Blundell, T L** *J. Appl. Crystallogr.* No 16 (1983)
3 **Peterson, M R and Poirier, R A** University of Toronto, Ontario, Canada
4 **Fletcher, M** *Comput. J.* No 13 (1970)

3

'An interactive graphics technique for colouring molecules' **Carlson, C N and Bosshard, H E** European Molecular Biology Laboratory, Postfach 10 2209, 6900 Heidelberg, FRG

Structural colouring serves to orient the viewer of a molecular model. It is an effective way of tagging features of interest so that they are recognizable in a jumble of bonds and electron densities. We have developed a technique of interactive colouring and implemented it as an addition to our version of the crystallographic program Frodo. Our foremost design goal was to offer as wide a range of functionality as possible, while keeping user interaction obvious and intuitive.

We designed a 'paintbrush' technique for colouring molecules which is like an artist dabbing a brush in the palette and painting with it. When the user enters 'colour' mode, a menu of colour options and a colour palette appear on the screen next to the molecular display (Figure 1). The user can at any time 'dip' a digitizing tablet stylus in the palette, picking up a colour whose value is monitored at lower screen right.

Subsequent identification of an atom with the stylus causes a group of atoms to assume the current stylus colour. The atoms belonging to the group are determined by the currently selected menu items. If, for example, the stylus colour is red and the user selects

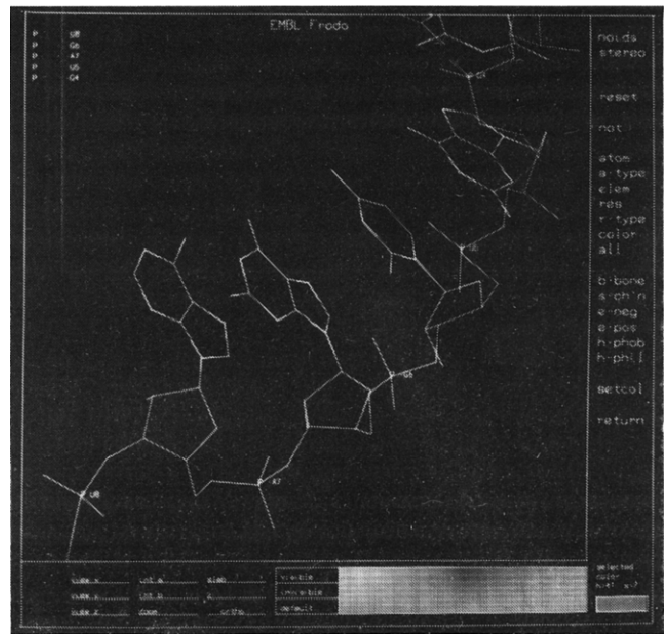

Figure 1. Molecular display in 'colour mode'

menu item 'a-type' (colour by atom type), then identifies an atom named 'CA', all atoms of the type 'CA' (carbon alpha) are coloured red. Using 'elem' (colour by chemical element) instead, all carbon atoms are coloured. In this scheme the order of stylus operations is unimportant, which keeps usage and implementation straightforwward. Most menu items remain selected after a colour operation so that multiple items (for example, multiple residues), can be coloured conveniently.

The menu provides for colouring the groups of atoms in Table 1.

An additional menu item, 'not', is used in combination with the menu in Table 1 to colour all except the selected group of atoms. Thus with 'not' and 'r-type' selected, identifying an atom in an alanine residue causes all displayed atoms except those in alanine residues to be coloured.

Experience with users quickly pointed up deficiencies in the colour menu and its usage. Users often wanted to colour one group of atoms the same shade as another, but had to rely on their eye for a good match. The 'setcol' item was added, with which the stylus colour can be picked directly from the molecule. A minor modification with a big impact on user comfort and security was the addition of a '... working' cue, which signals a colour operation in progress. Current implementation response can be slow for large structures (because of necessary disc accesses) and had often left users unsure if their stylus operations were being noticed.

The colours 'visible' and 'invisible', which are selected from boxes to the left of the palette, have proved particularly useful. Congested areas of a display can be cleared quickly by making undesired residues invisible. Backbone displays can be produced with 'not', 'b-bone', and 'invisible', and then sidechains added in areas of interest with 'res' and 'visible'. Clever colour partitioning of the molecule can be used to control the visibility of groups of atoms not directly provided for by the menu. For example, by giving the P–O5′–C5′–C4′–C3′–O3′ 'spine' of a nucleic acid a unique colour, the sugar rings and bases can be turned on or off by colouring 'visible' or 'invisible' everything which is not the spine colour.

Table 1. Groups of atom that can be coloured by the menu

Menu item	Atoms coloured
Atom	Identified atom
a-type	Atoms of the identified atom's type, eg 'CA' (carbon alpha) atoms
Elem	Atoms of the identified atom's chemical element, eg 'C' (carbon) atoms
Res.	Residue containing the identified atom
r-type	Residues of same type as the residue containing identified atom, eg 'ALA' (alanine) residues
Colour	Atoms of the same colour as the identified atom
All	All displayed atoms
b-bone*	Backbone atoms
s-ch'n*	Sidechain atoms
e-neg*	Atoms whose electro-negativity exceeds a predetermined threshold
e-pos*	Atoms whose electro-positivity exceeds a predetermined threshold
h-phob*	Atoms belonging to hydrophobic sidechains
h-phil*	Atoms belonging to hydrophilic sidechains

* These items are not yet functional in our implementation. The required information concerning charge, hydrophobicity, and backbone/sidechain membership must be added to the residue dictionaries.

Our implementation of interactive colouring allows manipulation of the colour and visibility of atoms. A straightforward extension could enable modification of further display qualities such as line texture, blinking, and brightness. The ability of the viewer to use these display qualities to impose his structural concept upon a displayed model, and to modify the representation interactively as his thinking progresses, is among the strengths of interactive graphics as a model-building tool.

4

'Computer animation as an aid to chemical understanding' **Crennell, K M, Crisp, G M and Dent Glasser, L S***
Computing Division, Rutherford Appleton Laboratory, SERC. *Chemistry Department, University of Aberdeen, Scotland

Computer Graphics is slowly displacing static mechanical models in popularity as an aid to the better understanding of chemical structure. We had no display devices working fast enough to show realistic moving structures and reactions, so we decided to use computer animation on film.

The SERC FR80 microfilm recorder was used to make a 16 mm colour film, 'Silicates in solution', which shows some of the possible mechanisms involved when silicates encounter water molecules in solution. Our approach to the problems of production is discussed, some typical computing times given and some still sequences from the film displayed.

Existing computer programs were used as far as possible; most of the moving sequences were made using PLUTO78, but we found it necessary to write a new one, TETRA, to simplify the manipulation of the shaded tetrahedra shown in the latter part of the film.

5

'Molecular interactive display and simulation: MIDAS' **Ferrin, T E, Huang, C, Jarvis, L and Langridge, R**
Computer Graphics Laboratory, University of California, CA 94143 , USA

MIDAS (Molecular Interactive Display and Simulation) is a large interactive molecular modelling graphics package developed at UCSF using the Bell Laboratories UNIX operating system. The system provides a flexible tool for the study of small and large molecules and their interactions, taking full advantage of available interactive 3D colour display capabilities. Bond rotation, interactive monitoring of several distances and 'docking' with realtime representations of van der Waals surfaces is well supported. Among its more innovative features is an unusually coherent hierarchical database for storage of macromolecules which minimizes both storage space requirements and access time. The 'tool building' philosophy provided by the UNIX operating system has resulted in a well-organized and maintainable program that is well suited to re-implementation on graphical workstations. Supported by research grant RR-1081 and a Guggenheim Fellowship (RL).