exploration and visualization of structures and properties. It should also allow the addition and modification of basic assumptions, inclusion of expertise and hypothesis, and handling of uncertainties to allow reasoning about unknown structures and their interactions. The Winchester Graphics System, developed at the UK Scientific Centre, allows great freedom in data manipulation, choice of graphical representation and output device. The system enables the user to pose queries about known molecular structures, and see the results graphically, with great ease and flexibility. Central to this flexibility has been the choice of a relational database; this allows data describing known structures and systems to be manipulated very easily. However, in the case of unknown structures and illunderstood systems, the user needs to be able to explore different approaches, assumptions and hypotheses without having to write new application programs every time. Experiments are under way to explore the use of the logic-based language PROLOG to augment the relational database, as a first step towards allowing the user easily to make and test hypotheses about unknown structures. Relations describing molecular structure are of three types: *general relations* contains information pertinent to all molecules, such as information about atom types, van der Waals radii, hydrophobicity, charge and amino-acid and nucleotide connectivities. *Specific relations* contain information describing a specific molecule, for example atomic identifiers and coordinates and residue types. *Derived relations* contain the information required for display. This is usually in the form of positional coordinates defining ends of chemical bonds and atomic positions. There are three types of statement in PROLOG: *Assertions* state facts. Molecule data are held as PROLOG assertions; organization of the data follows the relational database schema already in use. *Rules* take the form *x* is true if *y* is true. *Goals* ask questions. Preliminary experiments include:

- holding and querying protein data;
- construction of display lists for graphics devices;
- protein sequence searching;
- secondary structure prediction.

We are also exploring the use of PROLOG in the following steps which might be involved in the prediction of an unknown structure.

- Define a database of related or relevant known structures.
- Search for sequences matching the amino-acid or DNA sequence of the unknown structure. The match may be exact, may conserve residue volume, charge or hydrophobicity, or may minimize base-change-per-codon. The target sequence length may vary to optimise matching.
- Select and draw likely structures for interactive fitting to produce a plausible model structure, reassessing the likelihood of each structure in the light of others found.

18

Secondary structure determination as a starting point for tertiary modelization

J F Gibrat*, J Levin*, B Fishleigh†, B Robson*† and J Garnier*

*Laboratoire de Biochimie Physique INRA, Université de Paris-Sud, 91405 Orsay Cedex, France
†Theoretical Biochemistry Laboratory, University of Manchester, Manchester M60 1QD, UK

As the number of known protein structures deduced by X-ray crystallography has increased there has been a corresponding increase in the information available to make secondary predictions using unknown proteins. This information has allowed us to improve predictions based on the theory of information, as well as to develop new methods. These include predictions based on the patterns of hydrophobic and hydrophylic residues, and a prediction which searches for short homologous fragments of a test protein in a database of known proteins. These have been combined together to form an expert system capable of rapid determination of a starting conformation for further modelization. This program which requires no advance information about the type of protein, gives an average of 62% accuracy over 3 stages (coil, helix and β-sheet) on a database of 60 proteins and a reasonable certainty of finding all the secondary structure features within the protein, albeit a little displaced. This information can then be used to determine the type of super secondary structure motifs considered as a characteristic linear sequence of (a) helices and (b) β-sheets separated by connecting segments (X) leading to specific tertiary structures, such a Rossman folds (babab) or β-sheet barrels etc.... encountered in various known protein strutctures. By way of an important example of the use of secondary structure prediction as a starting point for tertiary structure calculation an interim prediction of the structure of chloramphenicol acetyl transferase will be presented.

19

Agraph: a program to build up animation films

Béatrice Amerein and Raymond Ripp

Laboratoire de Cristallographie Biologique, Institut de Biologie Moléculaire et Cellulaire, 15 rue René Descartes, 67084 Strasbourg Cedex, France
and
Départment Informatique de l'Université Louis Pasteur, 7 rue René Descartes, 67084 Strasbourg Cedex, France

Agraph is a program which produces animation films. It is designed to run on a Evans & Sutherland PS300. PS300 is a vectorized 3D graphic system which allows the manipulation, in real time, of complex graphic objects. Production of an animation logical sequence with Agraph is actually an interactive process using the PS300, based on the key scenes of the film to be made. The status of the actors in these scenes are stored on the host computer, while the pictures are not. The intermediate status of the actors are interpolated by the computer itself. The logical sequence built in this way can be displayed in real time; it can be improved by inserting or deleting other key scenes in an iterative process. When a real film is shot a producer tells the actors the main features of their play and corrects them as the rehearsal proceeds. This classical approach has also been chosen in Agraph to produce a sequence. Each scene is made out of a series of 3D graphic objects called 'actors' which are organized in a structured way. An actor may itself be a set of other actors. An actor is defined by its 3D position, its orientation, size and colour. In a key scene, these parameters are set to a given value. This is easily achieved on the PS300 using the interactive dials. The parameters which characterize the movement and the tuning of a fictitious camera are also stored and interpolated on the host computer. These are used to simulate the classical shooting techniques such as travelling and zooming. The interpolated actors and camera movements are continuous, giving smooth motions. Agraph is an easy-to-use program. It does not require computing knowledge from the operator. The production stage is carried out by the PS300 where each actor is defined in the following way:

```
ACTOR: = begin_structure
                 set contrast K ;
                 set color C ;
                 translate by T ;
                 scale by S ;
                 rotate by R ;
                 instance of X ;
            end_structure ;
```

where X could be a single object or a set of actors. Each actor is directed in the production stage : all his key status which will define the frame of his play are set up. Defining a key status means assigning values to *K, C, T, S* and *R*. These values are adjustable from the interactive dials. A key scene is the sum, for all actors, of their key status. Each of them is stored on the host computer. The different key status issued from the production stage and memorized on the host computer are interpolated. For each actor, interpolation by the host computer of his key positions produces a trajectory which is continuous and derivable. The trajectory goes through the initial positions: this is an important aspect as these positions may be meeting points. An axis and a rotation angle correspond to each orientation in the trajectory: these are also continuously interpolated as are the contrast, the colour and the scaling factor. This set of interpolated status which is actually the play of the actor is sent to the PS300. The animation is produced by assigning, in real time and to all actors, the status corresponding to their roles. The production is made in 3D space where the play occurs. The actors are filmed by an imaginary camera which can be moved, oriented and tuned. In order to achieve these adjustments, one has to define the line and angle of sight, the position of the clipping planes and the contrast value of the scene. These parameters are adjusted using the PS300 dials, during the production stage or even when the film is displayed. They are stored by the host computer and may be interpolated and sent back to the PS300 in the same way as the roles. The movement of the camera will then occur without any manual intervention. The film created in the way described above takes place on the PS300 screen. It may be interrupted at any time. This facility may be used to modify the scene displayed at this time by correcting, for example, the trajectory of an actor or the tuning of the camera. A new actor can also be added to an already existing sequence. His key status will be defined during the interruptions. Finally, two different sequences produced independently can be merged. It only requires the definition of the synchronization points. As a result of this iterative process, the new sequence can also be improved. This program was specially adapted to molecular graphics. A film on the structure of tRNA aspartic acid using Agraph will be shown.

20

Unix and the future of molecular graphics

T E Ferrin, C C Huang, G S Couch, K C R C Arnold and R Langridge

Computer Graphics Laboratory, University of California, San Francisco, CA 94143-0446, USA

Unix began as the efforts of one person in an industrial research laboratory to provide a more flexible and powerful operating system than those provided by the vendors[1]. It is now the most widely used operating system in the world, available on over 100 machines from micros to supercomputers, and is the system of choice in the overwhelming majority of computer science departments, both in the USA and elsewhere. In 1976 we chose Unix as the operating system for the UCSF Computer Graphics Laboratory using the new PS2/PDP-11/70 machines. A completely new graphics package was written for the PS2 under Unix, using C.[2] All program and system development at UCSF is done under Unix and molecular graphics research at UCSF is very effective in large measure because of the ease and flexibility which Unix provides[3,4]. Ports of our software to other operating systems are done only at the request of outside users (e.g. EMBL), and although these are very successful, there is a fair amount of criticism from others in the molecular graphics field on our choice of Unix. However, while most of our research and development until recently concentrated on DEC equipment (which is still our choice for most of our work) Unix allows portability and flexibility far greater than we could have hoped for at the time we adopted it. Unix-based systems have proliferated from the original DEC machines to virtually all vendors, not only to the ubiquitous workstations (Silicon Graphics Iris, Sun and even the IBM RT PC), and to the mid-range 'Crayettes' (Convex C-1), but even to a supercomputer (Cray 2). The original PS2 MIDAS system running on the PDP-11/70 proved very reliable, but it was the upgrade to the VAX 11/750 in 1981 which convincingly demonstrated the wisdom of choosing an operating system common to a variety of hardware. The transfer took only a few days. Since then we have made a similar port to the Silicon Graphics Iris 2400 and 2400T, and are investigating the Sun-3. We will describe our experience with Unix-based molecular graphics and outline the advantages of a standardized operating system for future development of the field, particularly in view of the inevitable trend to networked distributed workstation environments.