SQUID: A program for the analysis and display of data from crystallography and molecular dynamics

Thomas J. Oldfield

Department of Chemistry, York University, York, UK

SQUID is a flexible computer program that allows the analysis and display of molecular coordinates from crystallography, NMR, and molecular dynamics. The program can also display two-dimensional and three-dimensional data using many graph types, as well as perform array processing of data with numerous intrinsic functions. Graphics are based on the use of "move" and "draw" instructions, allowing easy development of new device drivers, including vector plotters.

Keywords: graphics, molecular dynamics, analysis, data processing, program

INTRODUCTION

A number of advanced graphical programs can be used for the display and manipulation of macromolecules (e.g., Frodo[1] and Hydra[2]). They allow density fitting, structure building, modeling, and the representation of the coordinates in many different ways. All of these programs require an advanced graphics machine for three-dimensional (3D) image manipulation in real time.

There were three main objectives for the program that make it different from other advanced programs:

(1) The ability to display information as two-dimensional (2D) and 3D graphs as well as atomic coordinates.
(2) Data processing facilities to enhance information from any file of data, as well as the possibility to add text labels and markers.
(3) The ability to produce the output on simple graphical devices, such as tektronics terminals or vector plots.

The last objective means the program can be used by the majority of people in the fields of crystallography and molecular simulation. SQUID can even be run on a standard VT100 terminal, allowing the user to view an approximation to the final graph, and hence plot the results. This can prevent much time wasted booking the "graphics" and then finding that the data were unsuitable.

The program has been designed to allow a user to study molecular coordinates, molecular dynamics (MD) trajectories, and 2D and 3D arrays of numbers as a series of pictorial representations. SQUID is therefore normally run as an interactive graphics program, where the user can browse through the different commands on subsets of the data. In this way, it is possible to correlate information graphically using data from various sources. At all stages, it is possible to annotate pictures with text and geometric shapes or to plot exactly the current display.

On starting the program, the initial parameter set is read in from a free format file, which the user can edit. The font and allowed command list are also read in, and the help file initialized. The user is then prompted for a protein data bank (PDB) or Karplus format coordinate file. (Either format is read automatically.) The user can also type *graphs* to indicate only the 2D/3D graph section is to be used, or *Gromos* to ask for a Gromos format coordinate file. (If just the *graphs* and *calculate* facilities are required, then SQUID allows only a subset of commands without the preceding *graphs* or *calculate* keywords.) After reading a coordinate file, the user is presented with the SQUID prompt and can access all the available commands in the main program loop. For ease of use, the program is divided internally and externally into subsections depending on the function of the commands. These divisions are shown in Figure 1. The eight sections are: *main, select, find, analysis, graphs, calculate, text* and *help*. The commands for *main* are issued directly to the Squid prompt, while keywords from other sections will only be recognized if preceded with the relevant section keyword (e.g., *graphs* or *find*). Each keyword normally describes the action to be taken, and can be shortened to three or four letters. For example, to draw a plot of all the phi and psi angles in a protein as a Ramachandran plot with the allowed region marked on:

Squid > find Ramachandran allowed_regions

Address reprint requests to Dr. Oldfield at the Department of Chemistry, York University, Heslington, York, Y01 5DD, UK.
Received 19 November 1991; accepted 11 February 1992

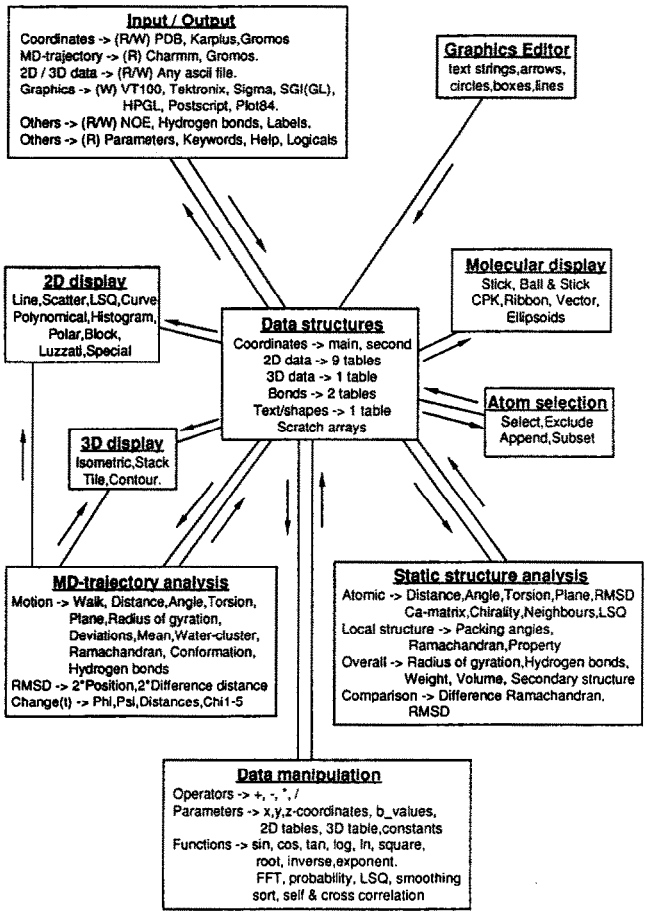

Figure 1. Program architecture for Squid.

or

Squid > fin ram all

The graphs section has multiple keywords that can affect the plot style. Consider the example:

Squid > graph file file overlay limits −2 2 noline symbol curve lsq

The program will prompt for two files of *xy* data to draw. The two sets of data will be drawn using the same axes (overlayed) with the *y*-axis limits between −2 and +2. *Noline* will prevent the default of joining each point with a straight line. *Symbol* will place two different symbol types at each point for the two graphs, while the *curve* and *lsq* keywords result in a cubic spline curve and a best-fit line added for each set of data. The data from the two files will fill the first two internal tables, and will be retained between commands until explicitly overwritten.

The program has several useful features to make the cumbersome keyword input easier to use. Command recall and editing is possible for repetitive operations on data. Commands can be streamed from an external file from within the program (e.g., for a standard set of text labels), or the whole program can be run in batch with no graphics produced. SQUID brings together many ideas for the analysis of proteins, such as Ramachandran maps and hydrophobicity profiles.[5] It also incorporates some novel ideas developed at York for the study of crystal structures and MD trajectories. These include the display of different Ramachandran plots, and graphical displays for RMSD in bonds and angles for proteins from crystallographic structures. The MD analysis allows the calculation of the angle and separation of secondary structure elements as a function of time, and a new way to display hydrogen bond information from a trajectory.

GRAPHICS AND PORTABILITY

There is currently a problem with graphics drivers for software, as there exist no standards. Squid was therefore designed so that a new driver can be written with very little difficulty. All plotting is carried out using the graphics primitives to draw a line and move to a point, including the text and hidden-line removal. The production of hard copy results are therefore possible on any type of plotter. The program currently has 9 available drivers for plotters and terminals, including an X-window version. The X-window version, which is still under development, uses all the versatility of the X-window buttons and menus but still uses the main functional code with no changes.

The program consists of approximately 30,000 lines of FORTRAN 77 code, which has been ported to the following computers successfully: VAX, Aliant, Convex, Silicon Graphics International, and Sun. In the case of the X-window version, the control section of the program is written in standard C.

SUMMARY

There are 215 possible commands in SQUID, each with a different set of parameters. It would be impossible to indicate the action of all the commands, so the following section summarizes the action of each program subsection. Finally, two simple examples of the use of SQUID are included.

Figure 1 shows a schematic of the program architecture with some of the possible commands listed. The central core of the program consists of several data tables that can be operated on by mathematical and logical operations, and copied to each other. Coordinate data can be used within the static structure analysis section of the program, or used as a template file within the molecular dynamics trajectory analysis.

The *main* head (molecular display) of the program is associated with the display of atomic coordinates, as well as with overall program control. The possible commands in the *main* head of the program are not particularly comprehensive, as they compete directly with the much more powerful programs available for atomic display, and of course it is not possible to emulate such programs on a tektronix, or X-window display. The program can produce basic mono and stereo pictures of stick, ball-and-stick and ribbon representations of molecules.

The *select* section of the program allows a subset of atomic coordinates to be used elsewhere in the program. The atoms selected are used within the *find* and *analysis* program subsections, allowing the user to focus, if necessary, on just a few atoms. The selection of atoms can be based on atom name, residue name, index number, sequence number, segment identifier, and a spherical region. It is possible to use

logical AND, OR, and NOT subselections for the atoms, allowing a large list of possible atom combinations.

The *find* section for static structure analysis is associated with the study of geometrical information for the current selection of atoms in the main coordinate table. This section of the program is of most use to crystallographers who are interested in the quality of refined coordinates, and who wish to display the properties of the coordinates in an informative way. Different functions are possible, ranging from the simple calculation of interatomic distance, to a complete table of secondary structure based on up to seven structural types.

The *analysis* section is used as a testbed for ideas on the methods used to analyze the extremely large amounts of data obtained from MD. The analysis functions consist of a set of commands that allow primary data reduction of transient coordinate information obtained from MD. Many of the simple commands involve the calculation of geometry (possible in the *find* section of the program) as a function of time. More strictly, this is a function of transient coordinate frames. The user can define a starting point, ending point and time step from a trajectory, and can also chain multiple trajectory files together. Possible analysis features as a function of time include the calculation of inter helix (or β-sheet) angles, Ramachandran plots, and hydrogen bond statistics. Although there are numerous possibilities for the primary analysis of dynamics information from proteins, the main advantages of the program SQUID are the display possibilities and data processing in the *graphs* section of the program.

The *graphs* routines form the largest and probably the most useful section of SQUID. In fact, the *graphs* section can be used as a stand-alone program for those people interested in displaying their own data. It is possible to display *xy* and *xyz* data using several different plot styles. The program supports: line, scatter, least-squares polynomials from the first to the eighth power, "perfect curves," histograms, block histograms, and polar graphs for *xy* data. For 3D data it is possible to draw isometric, 3D histograms (stack), contour, tile, and "balloon" plots. A second part of the *graphs* section is the *calculate* facility, which allows processing of data, and the transfer of information from one table type to another. The most obvious calculate command is the construction:

array = array .operator. array/constant

It is possible to use any of the main data tables within the command, while the four main mathematical operators are supported. A fifth possible operator is the @ or correlation function. The correlation operator allows the determination of self- and cross-correlation functions to be calculated between any set of data held in the 2D tables. Other possible functions include: sin, cos, tan, log, ln, exp, inverse, sort, probability, fast fourier transforms, and smoothing functions. The idea is to allow the user to process data from the *analysis* or *find* sections of the program and examine information of particular interest.

The *text editor* is a graphics screen facility that allows the placing of graphics objects anywhere on a plot. It can be accessed throughout the program, and can support up to 1000 character strings, boxes, circles, arrows, and lines. There are two methods to allow the input of the graphics objects, definition by a keyworded command for use, in particular, with streamed input files, or by cursor input. The text can be defined using different character sizes, color (where appropriate), and orientation. It is also possible to include nonstandard characters, such as Greek symbols from the internal font. The shapes can be defined using differently colored lines and five different line styles. These graphical objects can be grouped together so that they can be manipulated as if they were a single composite object. The text editor allows the enhancement of a plot to give publication quality graphics using only a few simple commands.

Help with any program is imperative. As well as written documentation, SQUID has several hundred pages of on-line help. The help is arranged in a hierarchical structure similar to the VAX VMS help system.

EXAMPLES

Interleukin 1 β

The following shows the simple analysis of the geometry for a protein from the Protein Data Bank (Interleukin 1 β[3]).

PDB Karplus file > *1ilb.pdb* <return>

Squid> *exclude residue HOH* <return>

Squid> *find weight* <return>

→ 17170 daltons

The program returns an instant approximation to the atomic weight

Squid> *find chirality* <return>

A check to find if any residues are in a conformation associated with D amino acids. None are found here.

Squid> *find cis* <return>

→ 1 cis peptide link found
→ bond between residues 90 - 91

(Proline 91 is documented in the REMARKS section as a cis residue)

Squid> *find ramachandran allowed* <return>

This command results in the Ramachandra plot for Interleukin with an allowed "hard sphere" region marked on (Figure 2). An energy surface is also possible.

What> *plot* <return>

Squid> *select atoms ca n c* <return>

Squid> *find eigen 3-12 142-152* <return>

→ separation = 8.5 Å
→ internal angle = 52.3°
→ out of plane angle (torsion) = −16.5°

The command calculates the angle, separation, and torsion angle, between the best line through the backbone atoms selected for these two ranges of residues. The best line is defined by the eigenvector associated with the highest eigenvalue calculated from the moments of inertia matrix.

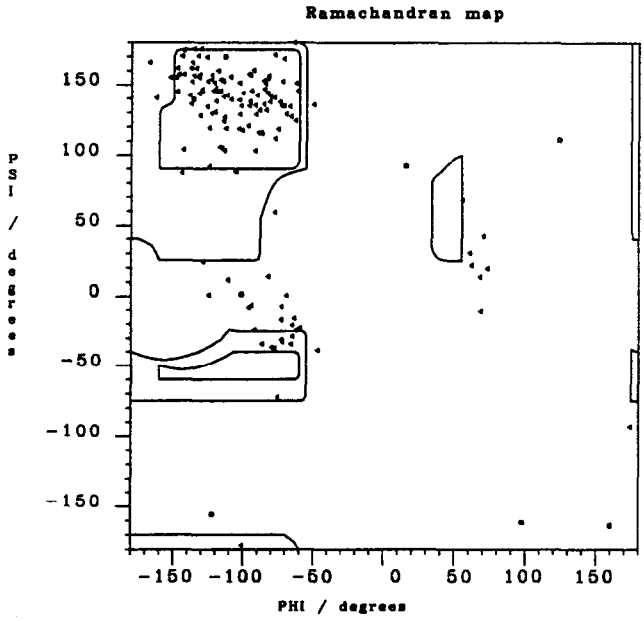

Figure 2. Ramachandran plot[3] for Interleukin 1 β with contours to mark the hard-sphere allowed regions. Glycine residues are marked with a circle and other residues with a triangle.

Squid> *select atoms ca* <return>

Squid> *link ca* <return>

Squid> *stereo* <return>

Squid> *ball_and_stick* <return>

The program uses the data in the *B*-value array to define the size of each ball in a plot. An overall scale factor in the parameter file sets up the scaling (Figure 3).

What> *plot* <return>

Squid> *mono* <return>

Squid > *graph calculate next = B*/26.3 <return>

The line calculates the expression as:

The next available graph table (which is table 1) is set equal to the scaled values in the *B*-value array of *haem.pdb*

Squid > *graph histogram* <return>

The data in table one is displayed on the graphics screen. The user is then prompted whether he wants to write a plotter file, or use the text editor. As the text editor is to be used:

What > *text* <return>

The cursor will appear at this point. The user moves the cursor using the cursor keys on the keyboard to vertical axis of the graph.

IAtom Fluctuation [9] [10] <return>

The Initial *I* starts the input mode for text.

This will place *Atomic Fluctuations (ASCII 9 ASCII 10)*, where: ASCII 9 = Å, and ASCII 10 = superscript 2.

R-90 <return>

To rotate the text by −90° clockwise so that the text lies up the axis

The next piece of text is required to label the bottom axis, so the cursor keys are used to get to this point

ISequence number <return>

E

The *E* ends the editing section and redraws the graph.

(Note that it is possible to replace the interactive text editing by the absolute text commands:

text define 30 40 "Atom Fluctuation [9] [10]" 2.5 1 −90.0

text define 60 20 "Sequence number"

The last three numbers define: Character size, Pen number, and Orientation, and are optional.)

What > *Plot* <return>

To plot the graph of atomic fluctuations for the Cα atoms (Figure 4).

Squid> *end*

Solvated lysine

The following series of commands will calculate the chi 1 and chi 2 angles of a solvated lysine residue as a function of time

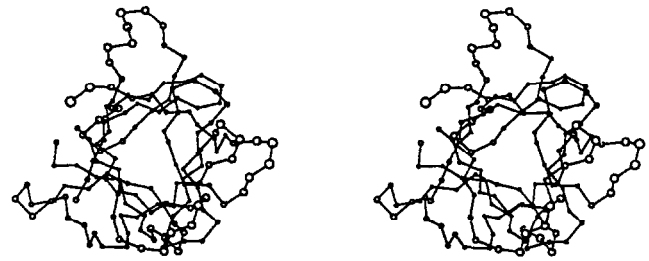

Figure 3. Stereo ball-and-stick diagram for Interleukin 1 β Cα atoms, where the radius of a ball is directly proportional to the temperature factor for each atom.

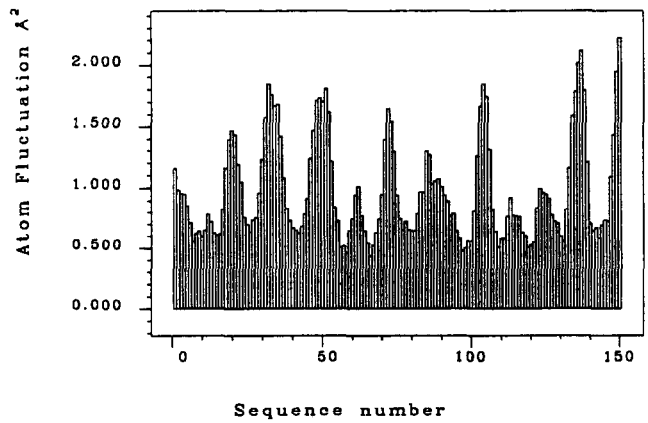

Figure 4. Histogram of the atomic fluctuations for the Cα atoms from Interleukin 1 β.

from a MD trajectory file. The simulation involved the analysis of the conformation of lysine 45 in porcine myoglobin using a stochastic boundary simulation at 900 K. To study the presence, or absence, of stable conformations, a 2D probability function is calculated for the two chi angles. Figures 4 and 5 show the results. This only shows a simple example of the possible analysis available. A comprehensive analysis of MD trajectories can be found in Oldfield.[4]

PDB/Karplus file >*myoglobin.crd* <*return*>

The file myoglobin.crd is a Karplus format file necessary to index the trajectory file.

Squid >*select sequence 45* <*return*>

The lysine atoms at sequence position 45 are selected.

Squid > *analysis torsion 45 n ca cb cg cd* <*return*>

The command is an analysis section command to calculate torsion angles. Notice that a single sequence number followed by five atom types will define two consecutive chi angles. The chi 1 and chi 2 values are returned in *xy* tables 1 and 2 (*g1* and *g2*).

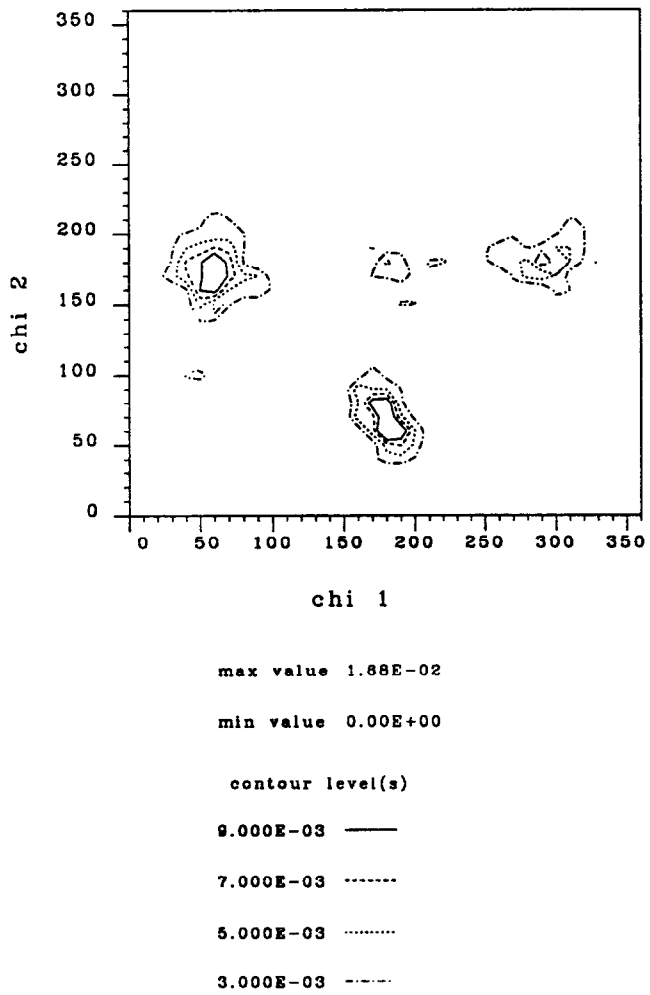

Figure 5. Contour plot for the probability function of chi 1 versus chi 2 from a MD simulation of a solvated lysine residue in porcine myoglobin.

trajectory file >*myoglobin.cor* <*return*>

options > *3000 100000 100* <*return*>

The trajectory file *myoglobin.cor* is a binary CHARMM format file. The options defined various parameters on file reading, the above sets the first frame of the trajectory at 3,000 femtoseconds, the last at 100,000 femtoseconds and a step of 100 femtoseconds.

The file will now be indexed and the torsion angles calculated for each frame. On reaching 100,000 femtoseconds, or the end of the file, the program prompts for another trajectory file. Pressing return results in the program drawing two separate *xy* line graphs on the screen of torsion angle as a function of time.

Another file ? > <*return*>

What > <*return*>

The following two commands change the range of angles over which the data is defined. Normally, the program returns all torsion angles between $-180°$ and 180°, but the angles of interest lie around 180°, on the edge of the plot. So define the torsion angle range between 0° and 360° for both sets of data in tables 1 and 2 (*g1* and *g2*).

Squid> *graph calculate postive g1* <*return*>

Squid> *graph calculate positive g2* <*return*>

Now generate a probability function of the data. The resulting probability function required is the 2D function, which should correlate the data in tables 1 and 2. (Note that 1D probability functions can be calculated with the command *graph calculate probability gn*, where *n* is 1 or 2.)

Squid> *graph calculate probability d(g1:g2)*
<*return*>

Each *y* value in table 1 will be "plotted" against the *y* value in table 2, and at the specified position, a bin will be incremented. We need to provide the bin sizes in *x* and *y*, so use 0°–360°.

xmin, xmax, and increment (default $-180,180,10$)

> *0 360 10* <*return*>

ymin, ymax, and increment (default $-180,180,10$)

> *0 360 10* <*return*>

Normalize the data (Y/N [N]) ? > *yes* <*return*>

Now label and draw the resulting 3D graph as a contour plot:

Squid> *Text define 60 20 "chi 1"*
Squid> *Text define 20 60 "chi 2" 2.5 1 −90*
Squid > *Graph contour*

Contour levels > *0.009 0.007 0.005 0.003*

Notice that the data is normalized, so that the total volume under the graph is one.

What > *plot* <*return*>

Now draw the data as an isometric graph.

Squid> *graph isometric*

Squid > *end* <*return*>

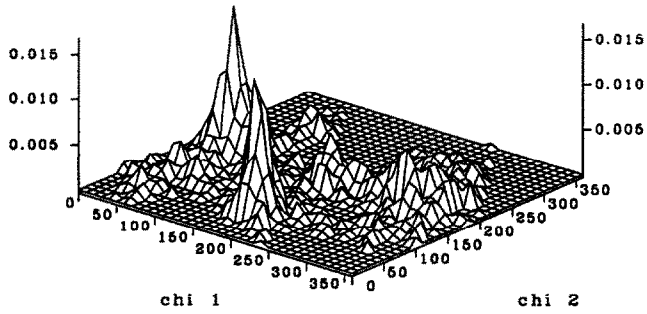

Figure 6. Isometric plot for the probability function of chi 1 versus chi 2 from a MD simulation of a solvated lysine residue in porcine myoglobin.

Figures 5 and 6 show the correlation of chi 1 to chi 2 in a solvated lysine residue during a simulation. The plot indicates that two major conformations for the lysine residue exist at different chi angles and these two angles are interdependent.

CONCLUSIONS

The program SQUID is a general graphical program for the analysis of information from molecular structures and MD simulations. It has a comprehensive data manipulation section for secondary analysis as well as the ability to display 2D data, 3D data and molecular coordinates in numerous ways. The code is available to academic institutions on request, while negotiations are in progress for marketing the program for nonacademic institutions.

ACKNOWLEDGMENTS

I thank Rod Hubbard and Guy Dodson for their encouragement and support. I am particularly indebted to Leo Caves, Andy Raine, and Chandra Verma for their many suggestions, criticism, and general comments on the program. Their constant help has resulted in an ever-decreasing number of bugs (and undocumented features), as well as many novel ideas within the program.

REFERENCES

1 Jones T.A. *Methods in Enzymology.* 1985, **115**, 157
2 Hubbard, R.E. *Current Communications in Molecular Biology: Computer Graphics and Molecular Modeling* (R. Fletterick and M. Zoller, Eds.) Cold Spring Harbor Laboratory Press, Cold Spring Harbor, 1986
3 Finzel, B.C., Clancy, L.L., Holland, D.R., Muchmore, S.W., Watenpaugh, K.D., and Einspahr, H.M. *J. Mol. Biol.* 1989, **209**, 779
4 Oldfield, T.J. PhD thesis, York University, 1990
5 Rose, G.D. and Dworkin, J.E. *Prediction of protein structure and the principles of protein conformation* (G.D. Fasman, Ed.) Plenum Press, New York, 1989, p. 625