POCKET: A computer graphics method for identifying and displaying protein cavities and their surrounding amino acids

David G. Levitt* and Leonard J. Banaszak†

Departments of Physiology and Biochemistry,† Medical School, University of Minnesota, Minneapolis, Minnesota, USA*

A new interactive graphics program is described that provides a quick and simple procedure for identifying, displaying, and manipulating the indentations, cavities, or holes in a known protein structure. These regions are defined as, e.g., the x_0, y_0, z_0 values at which a test sphere of radius r can be placed without touching the centers of any protein atoms, subject to the condition that there is some $x < x_0$ and some $x > x_0$ where the sphere does touch the protein atoms. The surfaces of these pockets are modeled using a modification of the marching cubes algorithm. This modification provides identification of each closed surface so that by "clicking" on any line of the surface, the entire surface can be selected. The surface can be displayed either as a line grid or as a solid surface. After the desired "pocket" has been selected, the amino acid residues and atoms that surround this pocket can be selected and displayed. The protein database that is input can have more than one protein "segment," allowing identification of the pockets at the interface between proteins. The use of the program is illustrated with several specific examples. The program is written in C and requires Silicon Graphics graphics routines.

Keywords: protein, enzyme, ducking, binding, surface

INTRODUCTION

The specific binding sites of proteins are normally located at indentations or cavities in the protein surface, allowing the protein to contact and recognize a significant fraction of the substrate surface. This paper describes a program for locating and investigating these "pocket" regions of protein. It takes as input a standard Protein Data Bank (PDB) file, and presents a three-dimensional (3D) display of the protein and all the pocket regions. The viewer can then interactively select one or more of the pockets and investigate them in detail. The surface of the pocket can be displayed as a transparent grid or as a solid. The protein residues that contact the surface can be selected and either highlighted, or displayed with the rest of protein removed. The program is written in C, and uses the GL (Silicon Graphics) graphics routines.

The procedure that most investigators now use to identify the pockets is based on the Connolly molecular surface (MS) algorithm.[1,2] This provides a very accurate description of the protein surface, which can then be used to find the pockets.[3,4] Recently, Ho and Marshall[5] have described a program specifically designed to visualize protein cavities. This program requires that one first specify a "seed" point in the cavity, and the bounding surface of this cavity is found by using solid-modeling techniques. One must have some initial idea about the location of the pocket to choose the seed point. In contrast, the emphasis of the new program described here is a quick procedure for routinely finding and investigating all the pockets, without requiring any prior information about their locations. Once the pocket and the residues surrounding it have been found, other graphics programs can be used, for example, to position substrates in the pocket.

PROCEDURE FOR IDENTIFYING THE POCKETS

The program uses a simple method for detecting pocket regions. For a given (fixed) value of y and z, a sphere of radius r is moved along the x axis in discrete steps (δ). At each x position, a test is made to determine whether the center of any protein atom is within this sphere, i.e., of distance less than r from (x,y,z). Regions are looked for, as one moves along x, where there is first, protein contact with the sphere, then no contact, then contact again. The region of no contact that is surrounded by contact regions is assigned a density = 1 (all space is initially assigned a density = 0). This x-axis scan is repeated for all y and z values. Then, the same procedure is repeated for y scans (x and z fixed) and z scans (x and y fixed). The pocket regions are defined as all points having a density

Address reprint requests to Dr. Levitt at the Department of Physiology, 6-255 Millard Hall, University of Minnesota, Minneapolis, Minnesota 55455, USA.
Received 19 July 1991; accepted 7 January 1992

value of 1. For all the results described here, a value for r of 3 Å was used.

The heart of the program, and the feature that distinguishes it from other programs (such as that of Voorintholt et al.[6]), is the algorithm that is used to display the pockets after the density map has been found. The surfaces of the different pockets are displayed using a variant of the marching cubes algorithm of Lorenson and Cline,[7] as modified by Knox.[8] The total volume is divided into cubes (of length δ on each side). The values of the density at the 8 cube vertices determine whether the surface passes through this cube; the shape of the surface is defined by a set of triangles for each cube. The algorithm had to be modified for this application because there will be many distinct closed surfaces, corresponding to the different pockets, and it is necessary to know which cubes belong to the same surface. This allows one to select a specific surface simply by clicking the mouse button when the pointer is touching any line of the surface.

The following procedure was used to assign to each cube the number of the surface that it belongs to. As one marches through the cubes in the usual way, if a cube vertex has a density = 1, then the other vertices of the cube are checked to see if they had previously been assigned to another surface (as a vertex of a neighboring cube). If none has been so assigned, the point is assumed to be part of a new surface and is assigned a new value. If one or more of the other points belong to another surface, then the new cube is assigned to the value of this surface. Since there will be lobes in the surface, it is necessary to account for the merging of what initially appear to be separate surfaces into a single surface. The merging of the surfaces is indicated by the other cube vertices having been previously assigned to two different surfaces. When this is encountered, all the cubes that had been previously assigned to the higher number surface are reassigned the value of the lower number surface.

The surfaces are modeled either as a gridwork of lines or as a solid surface. The solid surface requires determination of the outward normal to the surface. The GL graphics routines allow the use of the normal at each vertex of the triangle to smooth the surface. However, this was not done because it was felt that a simple planar surface (see the figures) provides a better description of the pocket. Given the set of triangles that form the closed surface along with their outward normals, the volume of the pocket can be determined analytically using the Divergence Theorem (Gauss's law) to convert from a surface (S) integral to a volume (V) integral:

$$\int_V \nabla \cdot u dv = \int_S u \cdot n ds$$

where u is an arbitrary vector and n is the surface normal. Choosing u equal to the vector x (the x component of the vector from the origin to the surface element):

$$\nabla \cdot u = 1; \qquad V = \int_S x \cdot n ds$$

Since the surface is made up of a set of triangles, the volume is found by summing the dot product of x and n over all the triangles, where x is the x component of the center of the triangle, and n is the outward normal of the triangle.

The accuracy of the surface depends on the step size (δ) that is used. If an infinitely small step size were used, then the surface determined by this algorithm would be exactly r (3.0 Å) away from the center of the closest (non-hydrogen) atom. The use of a finite δ means that the actual surface is more than 3.0 Å from the closest atom, with an average distance of about 3.0 + δ/2 Å. (For a δ of 2.0, the average distance is 4 Å, which is close to the van der Waals radius of unbonded atoms, when the hydrogen atom is included.) Since the position of the test grid points relative to the protein depends on the orientation of the protein, the larger the value of δ, the more the fine structure of the surface will depend on the initial protein orientation. Although decreasing δ increases the accuracy of the surface, one cannot use too small a value of δ because the time required to find the density is proportional to $(1/\delta)^3$. The size of the step that is used depends on the size of the pocket that one is looking for. If one is looking for a large pocket (such as are used in the examples below) then one could use a large δ (say, 2 Å), while using a δ of this size might miss a small pocket. There is probably no value in using a δ much smaller than 1 Å, since there is an inherent uncertainty in the position of the surface because hydrogen atoms are neglected and because the method uses the same value of r (3 Å) for all atoms. The program accurately identifies and displays all the amino acid residues and the atoms that surround the pocket, and these can be used to provide a more accurate description of the pocket. If one wants a high-resolution display of the van der Waals surface of the cavity, then other programs, more suitable for these tasks, should be used.[5] Even in these cases this program would be useful since it could be used to locate the regions of interest and serve as a first stage in the process.

PROCEDURES FOR DISPLAYING AND MANIPULATING THE PROTEIN AND SUBSTRATE STRUCTURE

The PDB file is read and the protein is displayed using standard procedures. A look-up table is used to determine how to connect the amino acid atoms. If there are substrate or prosthetic groups in the structure, their connectivity is determined by an algorithm that connects atoms whose interatomic distance is less than 2.0 Å. Waters are also read and displayed. However, the waters are not used in determining the position of the pocket. That is, it is assumed that the waters could be displaced by substrate and thus are not a fixed part of the protein structure. The hydrogen atoms are not displayed or used in the calculation. The PDB data base can contain 2 or more "segments," i.e., separate proteins. This allows identification of "pocket" regions in the interface between neighboring proteins (see Figures 6 and 7).

Once a particular surface has been selected, the amino acid residues that surround the pocket can be easily determined. A particular surface consists of a list of cubes that are intersected by the surface. All the atoms in the protein are tested to determine if they are within a certain contact distance of the vertices having a density of 1 in the cubes that form the surface. The contact distance that was chosen was the sum of the radius of the test sphere (r) plus the step size (δ), since, from the definition of the surface, there must be at least one contact atom within this distance. The program allows the

display of the amino acid residues containing atoms that contact the surface, the contacting atoms, or both.

COMPUTATIONAL DETAILS

The only part of the program that takes any significant amount of computer time is the calculation of the density map. Calculation of this map for a protein with 1017 atoms (131 amino acids) and a grid size (δ) of 2.0 Å (the value that is routinely used), has 9,500 grid points and requires about 2.5 minutes on a Silicon Graphics Iris 4D/25. The time is proportional to the $(1/\delta)^3$. Once the density file (which consists simply of a set of 1s and 0s) has been found, it is stored as a binary file. In subsequent runs, for example, to select a different pocket region, this density file is used and the program executes without any noticeable delay. As implemented, the program uses the SGI dial box (for zooming, and for rotation and translation about the x, y, and z axes) and button box (for selecting the various options). The program has also been implemented to use the CrystalEyes (StereoGraphics Corporation) stereo system. The black and white figures that are shown here were made by using the SGI program *icut* to capture the desired region of the screen, which was then printed on a postscript printer. In these printed figures, the background is white, while the normal graphics display is on a black background.

ILLUSTRATION OF THE USE OF THE PROGRAM POCKET

When the program is first run, it asks for the name of the PDB file and for the grid step size (δ) to use. (All the results displayed here use a default radius of 3 Å for the test sphere radius r.) The program then scales the data to fill the graphical screen and brings up a split screen 3D image of the molecule and the pockets. Figure 1 shows (in a black and white postscript copy of the screen) an example of this display for the adipocyte lipid binding protein (ALBP) whose structure has recently been determined in our laboratory.[9] This protein is one of a series of homologous proteins that are known to bind lipid in a pocket in the center of the molecule. Most of the pockets seen in Figure 1 represent shallow indentations in the surface. The image in Figure 1 has been rotated, revealing that one of the pockets corresponds to a large cavity in the center of the molecule, the lipid-binding region. Only the main chain atoms of the protein are shown. Figure 1 shows the screen after the pointer (arrow) has been placed on one line of this surface and clicked, selecting and highlighting this surface. Figure 2 shows this same screen using the solid surface option. It provides a clearer view of the spatial relation of the different pockets. Pressing the *Display Selected Pocket* button then removes all the other pockets, displays just this one (Figure 3), and prints the volume of the selected pocket at the bottom of the screen. Clicking on the selected pocket "deselects" it, and a different pocket can be chosen.

Once the pocket has been selected, the amino acid residues and the atoms that surround it can be found (by pressing a button). The selected pocket and associated residues are then saved, and they can be used immediately in subsequent runs. The program provides a large number of options for displaying the selected pocket and protein (chosen by using the SGI button box). In Figure 3, main chain atoms belonging to residues that contact the pocket are drawn in a different color (different line density in the black and white screen copy). In Figure 4, only those residues that contact the selected surface are displayed.

If there are substrates or prosthetic groups in the PDB file, they are also displayed. Figure 5 illustrates this with another protein whose structure has recently been determined in our

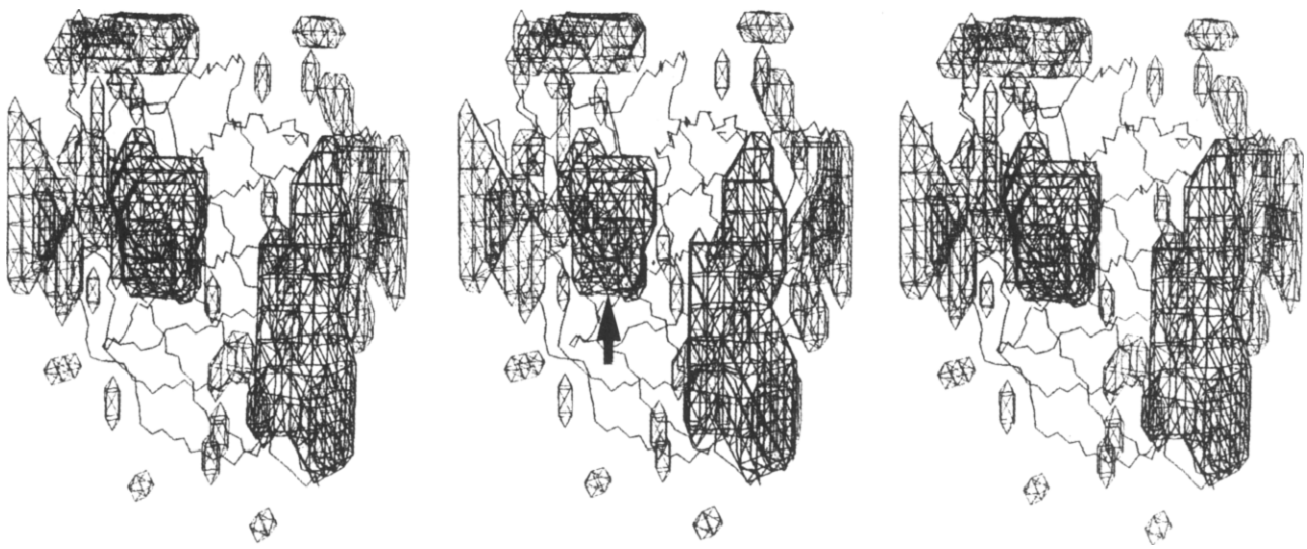

Figure 1. Black and white stereo display of all the "pockets" in ALBP. The left pair of images is for direct-eye viewing, and the right pair is for cross-eyed viewing. The image has been rotated so that the large cavity in the center of the molecule can be more easily seen. The arrow indicates where the pointer was placed and "clicked," selecting and highlighting the central cavity surface. The atoms that form the pockets cannot be seen because only the main chain atoms of the protein are displayed.

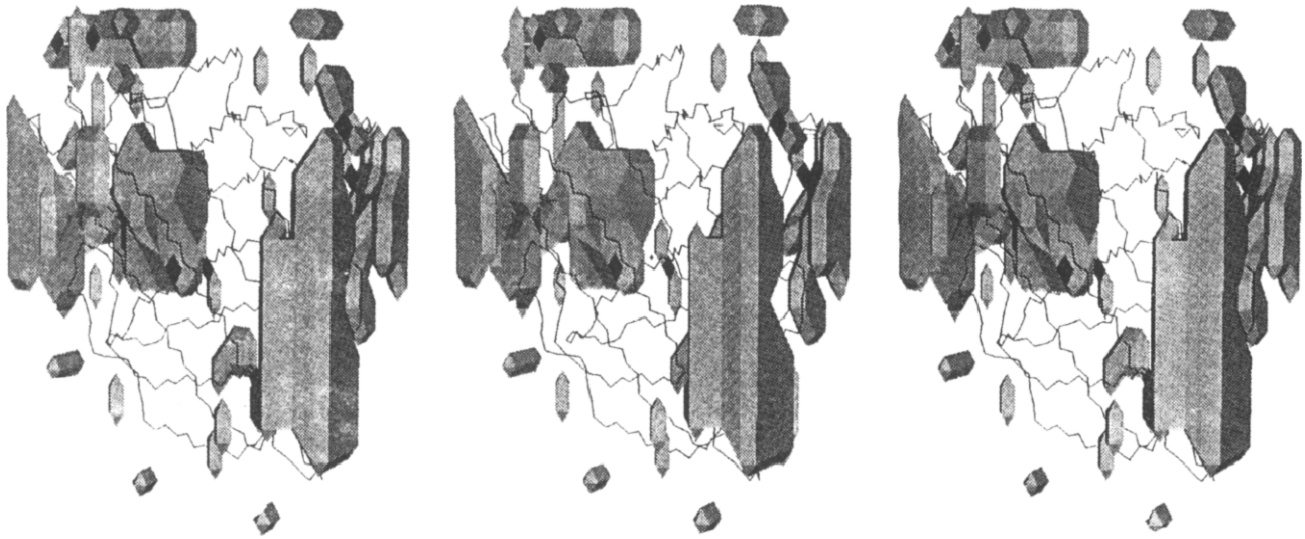

Figure 2. Similar to image in Figure 1, except that the "solid surface" option has been chosen using the SGI button box.

Figure 3. Stereo display of solid surface of just the "selected" pocket. The main chain atoms of the residues that contact the selected surface are displayed in a different color (less dense line in black and white). This is an option that can be selected using the button box.

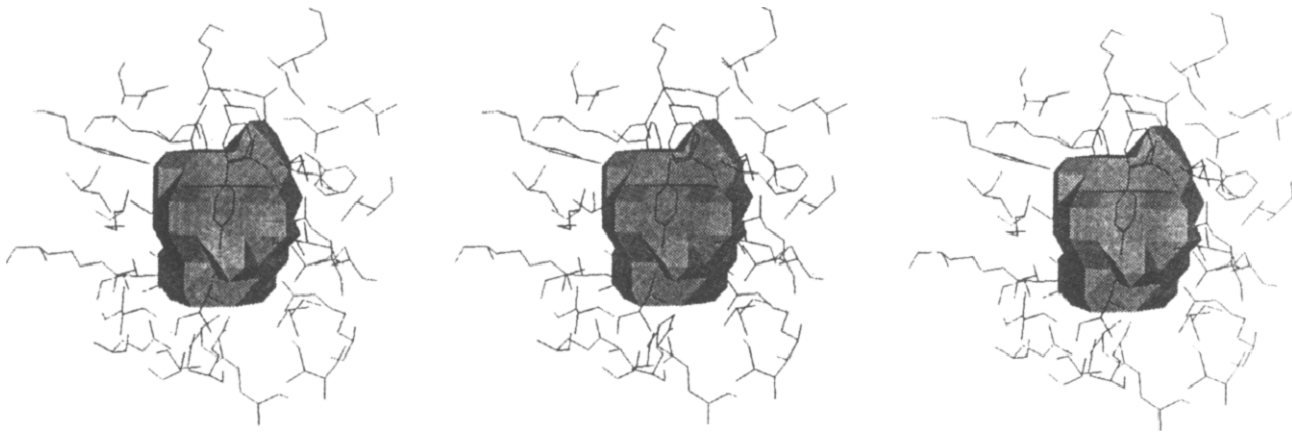

Figure 4. Stereo display of the solid surface of the selected pocket and the amino acids that contact this pocket.

Figure 5. Stereo display of the surface of the pocket in EMDH that corresponds to the NAD binding site. The position of the substrate malate is also shown.

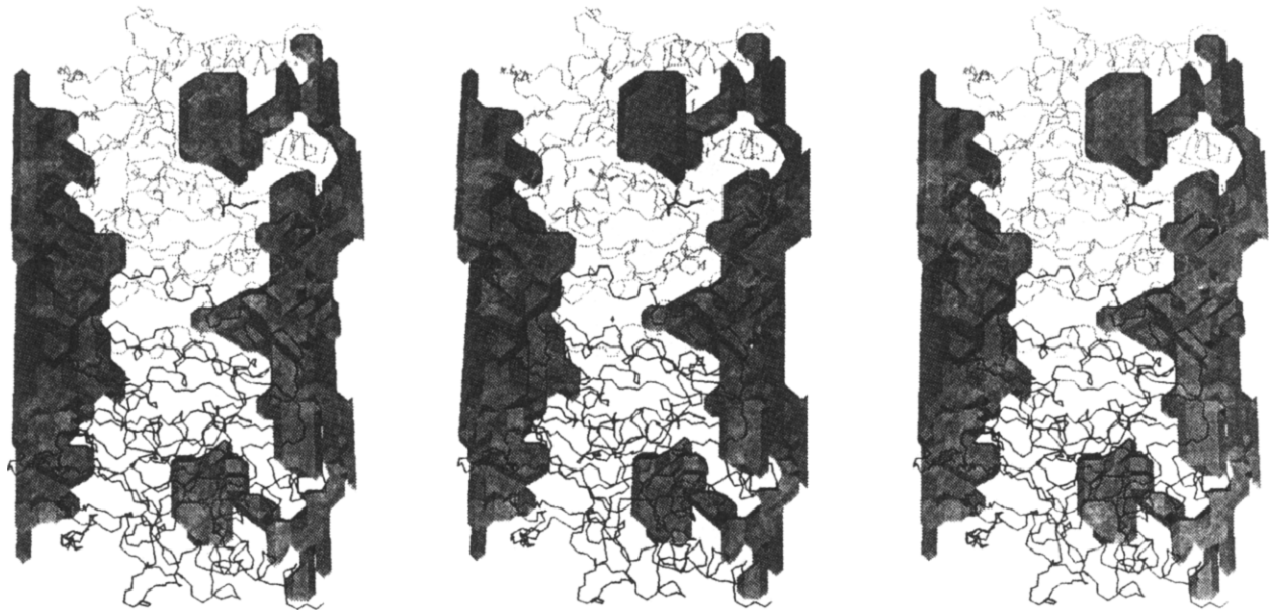

Figure 6. Stereo display of two selected pockets that show the region of contact of the EMDH dimer. (One EMDH molecule is drawn with less dense lines.) The peak in the right surface lies on a symmetry axis of the dimer.

Figure 7. Stereo display of the large pocket formed by the uteroglobin dimer. One monomer is drawn with a less dense line.

laboratory, *E. Coli malate dehydrogenase* (EMDH).[10] The malate is highlighted and labeled. (Labels can be placed on any residue by clicking the mouse button on them.) The large pocket next to the malate presumably corresponds to the region occupied by NAD, a cofactor for the enzyme.

The interface region between neighboring proteins can be investigated by including two or more proteins (as segments) in the PDB file. EMDH exists as a dimer in solution, and Figure 6 illustrates the use of POCKET to display the regions of tight contact in the dimer. The large extended pockets in Figure 6 show the indentations at the top and bottom of the contact region. There are no other pockets between these two. (That is, there is no region in which the 3-Å test sphere could be placed without contacting atoms from one of the proteins.) Figure 7 shows another application of POCKET to the dimer interface region. Uteroglobin (file 2UTG, Brookhaven Data Bank) is a dimeric protein that is believed to bind progesterone in a pocket formed by the dimers. The main chain of the two monomers and the pocket is displayed in Figure 7.

CONCLUSIONS

This program provides a simple tool for quickly and routinely locating all the indentations and cavities of a protein whose structure is known. Its main purpose is to find these pockets and identify the amino acid residues that surround them. A large number of options are then available for displaying a pocket and the protein. Once the pocket has been found, other programs can be used, e.g., to dock substrates in the pocket or to obtain a high-resolution image of the surface of the pocket.

The POCKET software can be obtained by contacting D.G.L. (via the Internet: levitt@dccc.med.umn.edu).

ACKNOWLEDGMENTS

We wish to thank Michael D. Hall and Zhaohui Xu for allowing us to use their unpublished protein structures, William Gleason for his suggestion that we look at the pocket in uteroglobin, and Charles Knox for his help with the marching cubes algorithm.

REFERENCES

1 Connolly, M.L. Analytical molecular surface calculation. *J. Appl. Crystallogr.* 1983, **16**, 548–558
2 Connolly, M.L. Solvent-accessible surfaces of proteins and nucleic acids. *Science* 1983, **221**, 709–713
3 Kuntz, I.D., Blaney, J.M., Oatley, S.J., Langridge, R., and Ferrin, T.E. A geometric approach to macromolecular-ligand interactions. *J. Mol. Biol.* 1982, **161**, 269–288
4 Tilton, R.F. Jr., Singh, U.C., Weiner, S.J., Connolly, M.L., Kuntz, I.D., and Kollman, P.A. Computational studies of the interaction of myoglobin and xenon. *J. Mol. Biol.* 1986, **192**, 443–456
5 Ho, C.M.W. and Marshall, G.R. Cavity search: An algorithm for the isolation and display of cavity-like binding regions. *J. Computer-Aided Mol. Design* 1990, **4**, 337–354
6 Voorintholt, R., Kosters, M.T., Vegter, G., Vriend, G., and Hol, W.G.H. A very fast program for visualizing protein surfaces, channels and cavities. *J. Mol. Graphics* 1989, **7**, 243–245
7 Lorensen, W.E. and Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. *Comp. Graphics* 1987, **21**, 163–169
8 Knox, C.K. and Wenstrom, J.C. 3D visualization of neural structures. In *Proceedings, Society for Computer Simulation. Eastern multi-conference*. Nashville (1990) 12–17
9 Xu, Z., Bernlohr, D.A., and Banaszak, L.J. The crystal structure of recombinant murine adipocyte lipid binding protein. *Biochemistry* 1992, **31**, 3484–3492
10 Hall, M.D., Levitt, D.G., and Banaszak, L.J. Crystal Structure of *Escherichia coli malate dehydrogenase*. *J. Mol. Biol.* 1992, **226**, 867–882