ELSEVIER

Visualizing protein conformational changes on a personal computer – alpha carbon pseudo bonding as a constraint for interpolation in internal coordinate space

Andrew G. Booth

School of Biochemistry & Molecular Biology and Flexible Learning Development Unit, University of Leeds, Leeds, UK

A Java program has been developed to produce animations and movies of protein conformation changes. The animations, primarily intended for teaching purposes, are produced by visualizing a series of aligned structures interpolated between two forms of the same protein. To produce plausible intermediate structures, the interpolation is performed in internal coordinate space and uses a simple constraint to avoid the production of artifactual movements of the protein backbone. This constraint involves the introduction of 'pseudo' bonds linking the protein alpha carbon atoms. All of the steps from alignment of initial structures to the production of AVI movies can be performed on readily available personal computers. © 2001 by Elsevier Science Inc.

Keywords: conformational change, animation, movies, interpolation, molecular modelling, proteins, personal computers

INTRODUCTION

The visualization of conformational change in proteins can be useful in understanding the mechanisms behind their biological functions. Animations of the movements involved are more easily comprehended than a series of static pictures and are particularly useful for teaching. Usually, such animations are produced by aligning the atomic coordinates of two forms of the same protein, the data usually being obtained from the Protein Data Bank.[1] Toggling images of the two structures (e.g., as kinemages[2]) can give an impression of conformational change but smoother animation can be achieved by inserting images of intermediate structures. These hypothetical intermediate structures (referred to in this study as 'intermers') are then visualized in sequence to produce the animation. The visualizations may take the form of digital movie files (e.g. AVI, Quicktime or MPEG) or the intermer coordinate files may be visualized (e.g., by using CHIME via Eric Martz' Protein Morpher. CHIME can be downloaded from: http://www.mdli.com/ and Eric Martz' Protein Morpher is at http://www.umass.edu/microbio/chime/morpher/).

The simplest method for producing the intermers is to linearly interpolate the atomic coordinates of the two extreme structures. This method, interpolation in Cartesian space, was used by Vonrhein et al.[3] to produce movies of the conformational changes produced by substrate binding to nucleoside monophosphate kinases. At the same time, members of the BioNet Project (part of the UK Teaching & Learning Technology Programme) were using the same technique to produce movies of oxy/deoxy-hemoglobin. (The animations produced by the BioNet TLTP project can be found at http://www.leeds.ac.uk/bionet/animation/mol_anim.htm.)

Interpolation in Cartesian space has severe limitations. The bond lengths and angles of the intermers will be unrealistic and several movies produced by this method clearly show protein chains passing through each other. To a certain extent, these problems can be overcome by applying suitable restraints and by energy-minimizing each intermer. This is the approach taken by Gerstein and Krebs,[4] who have set up a protein morph server at http://bioinfo.mbb.yale.edu/MolMovDB/morph/.

An alternative approach is to use interpolation in internal

School of Biochemistry & Molecular Biology and Flexible Learning Development Unit, University of Leeds, Leeds LS2 9JT, UK. Tel.: +44 133 322 4304. Fax: + 44 133 233 4713.

E-mail address: a.g.booth@leeds.ac.uk (A.G. Booth)

Journal of Molecular Graphics and Modelling 19, 481–486, 2001

655 Avenue of the Americas, New York, NY 10010

1093-3263/01/$–see front matter
PII S1093-3263(01)00088-7

coordinate space. In this method, it is not the atomic coordinates that are interpolated, but the bond lengths, bond angles, and torsion angles. The intermers are then constructed and aligned accordingly. This approach is used by Gerard Kleywegt's LSQMAN program.[5] Ideally, this method produces intermers with realistic bond lengths and angles. However, if there are substantial differences in backbone torsion angles between the start and end structures, the intermediate torsion angles will deflect the protein chain into distorted shapes and hence produce artifactually large movements. Even relatively small local distortions can become amplified as the intermers are assembled. Despite this potential problem, interpolation in internal coordinate space offers a reasonable compromise between the computationally undemanding but oversimplistic linear interpolation and more exact but computationally demanding methods such as those relying on molecular dynamics calculations. An example of the use of molecular dynamics for the production of molecular animations is provided by Jakob Halaska at: http://www.biokemi.su.se/~jakob/homepage/.

The aim of this study was to implement interpolation in internal coordinate space on personal computers that are readily available to teachers (e.g., a 166-MHz Pentium running Windows 95/98/NT or an Apple iMac) and to incorporate a simple constraint that will generally prevent the production of the artifactual distortions described above. The aim was simply to produce animations and movies containing 'plausible' structures that can be used for teaching purposes, not to implement algorithms that explore the actual conformational pathways by which one form of a protein changes to another such as the Stochastic Difference Equation[6,7] or the Conjugate Peak Refinement method.[8]

METHODS

Protein coordinate files were downloaded from the Protein Data Bank.[1] Structures were aligned using Swiss-PdbViewer (http://www.expasy.ch/spdbv/).[9] The algorithms for performing the interpolation in internal coordinate space were based on Jon Maber's unpublished C++ Interpol program originally written for the production of movies of small molecules for the BioNet Project. To ensure portability, the program developed during this study was written as a Java 1.1 application using the Java AWT for the user interface. This application—Java Interpol—reads the aligned coordinates files of the start and end structures in PDB format,[1] and calculates and interpolates all bond lengths, bond angles, and torsion angles. It also identifies the three atoms (X,Y, and Z) that move least between the start and end structures. Tables of bonded atom pairs, bond angle atom triplets, and torsion angle atom quadruplets are produced. The program then takes one of the atoms (A) from the first entry in the table of bonded atoms and places it at the origin of a Cartesian coordinate system. Its partner atom (B) is then placed on an axis, a distance from the origin equivalent to the A–B bond length. This pair of bonded atoms is then found in the table of bond angle triplets and the third partner atom (C) is identified. This is then positioned in the coordinate space as defined by the bond angle A–B–C and the bond length B–C. The triplet ABC is then found in the torsion angle quadruplet table and the fourth partner atom is identified (D). This is then positioned in the coordinate space as defined by the torsion angle A–B–C–D, the bond angle B–C–D, and the bond length C–D. The program then works its way through the table of bonded atoms, identifying those that have not yet been positioned and setting their positions as described above. The final structure is then aligned with the 'start' structure using as key the three atoms X,Y, and Z.

The program can produce the requested number of aligned intermediate structures either as separate PDB files or as a single NMR PDB file together with the CHIME scripts needed for animation. The current version of Java Interpol allows the introduction of extra 'pseudo bonds' into the interpolation (see below). During the assembly of intermers, the atoms linked by pseudo bonds are positioned before the positioning of those linked by conventional bonds.

In this study, the intermers in the NMR PDB files produced by Java Interpol were overlaid in Swiss-PdbViewer. Since Swiss-PdbViewer does not allow printing directly from the program, the images were saved as bitmaps and transferred to a graphics package (Paintshop Pro 5; Jasc Software, Inc., MN) for color adjustment and printing. Animations were produced in CHIME using the NMR PDB files, and in associated scripts produced by Java Interpol. Movie files in AVI format were created by producing individual PDB files for each intermer and, if necessary, energy-minimizing them using the Swiss-PdbViewer implementation of the GROMOS96 force field. The intermers were then visualized using the Win32 version of Mike Carson's Ribbons program[10] and the images produced were transferred to Paintshop Pro 5 for reduction of color depth to 8 bits per pixel before being combined into an AVI movie using Microsoft VidEdit 1.1. (VidEdit is no longer supported by Microsoft, but it remains one of the best programs for producing AVI movies. Version 1.1 can still be downloaded by anonymous FTP from: ftp.microsoft.com/developr/drg/Multimedia/Jumpstart/VfW11e/dk/winvideo/.)

RESULTS

Interpolation in Internal Coordinate Space of Human Hemoglobin

An initial examination was made of the results of interpolation between the deoxy- and oxy- forms of the alpha subunit of human hemoglobin. The PDB files used were 1HHO and 1BZ0. The aligned backbone structures are shown in Figure 1a. An unconstrained interpolation was performed using Java Interpol to produce ten intermers (i.e., 1HHO, 1BZ0, and eight intermediate structures). The overlaid structures are shown in Figure 1b. The intermers showed much more movement than would have been expected. In particular, there was a large reciprocating movement of the C helix (at the bottom of the overlay) and considerable movement of the A helix (at the top of the overlay). Both of these motions involved extensive movement of the backbone outside of the space between the start and end structures.

Introduction of Alpha Carbon Pseudo Bonds

Examination of the overlaid backbone structures of 1HHO and 1BZ0 showed that there are large differences in the backbone torsion angles involving the peptide carboxyl oxygen atoms at HIS50 and ALA71. The effect of these structural differences was examined by repeating the interpolation with a bond introduced between the alpha carbon atoms of ALA71 and HIS72 to 'bypass' the peptide bond. This 'pseudo bond' was treated as a real bond during the interpolation and construction of intermers, but was not included in the final structures. The Java

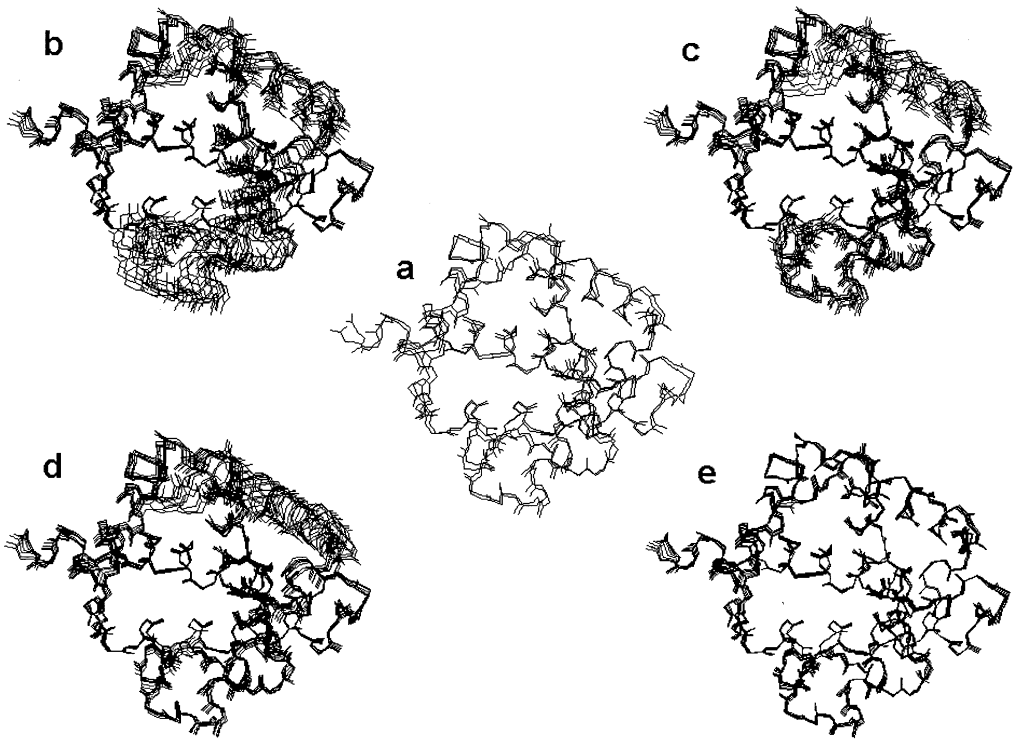

Figure 1. Interpolation of hemoglobin. (a) Superimposition of the aligned backbone structures of oxy and deoxy forms of the alpha subunit of human hemoglobin. (b) Superimposition of ten intermers produced by an unconstrained interpolation in internal coordinate space between the two structures shown in (a). Note the large movements of the A helix (top) and the C helix (bottom). (c) As (b), but constrained by the introduction of a pseudo bond between the alpha carbon atoms of ALA71 and HIS72. (d) As (c), but with an additional pseudo bond between the alpha carbon atoms of HIS50 and GLY51. Note that the constraints in (c) and (d) decrease the artifactual movement of the C helix, but have little or no effect on the movement of the A helix. (e) Result of interpolation with a three-level constraint of all of the alpha carbon atoms. All the intermediate structures now occupy the space between the two structures shown in (a).

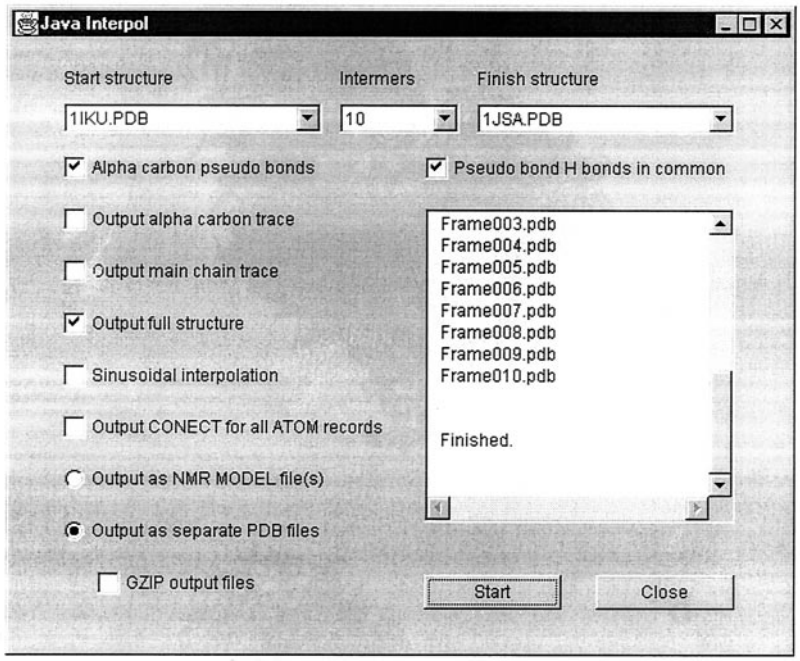

Figure 2. The user interface of Java Interpol. Additional constraint can be applied by pseudo bonding the alpha carbon atoms of residues involved in hydrogen bonds that are found in both the start and end structures.

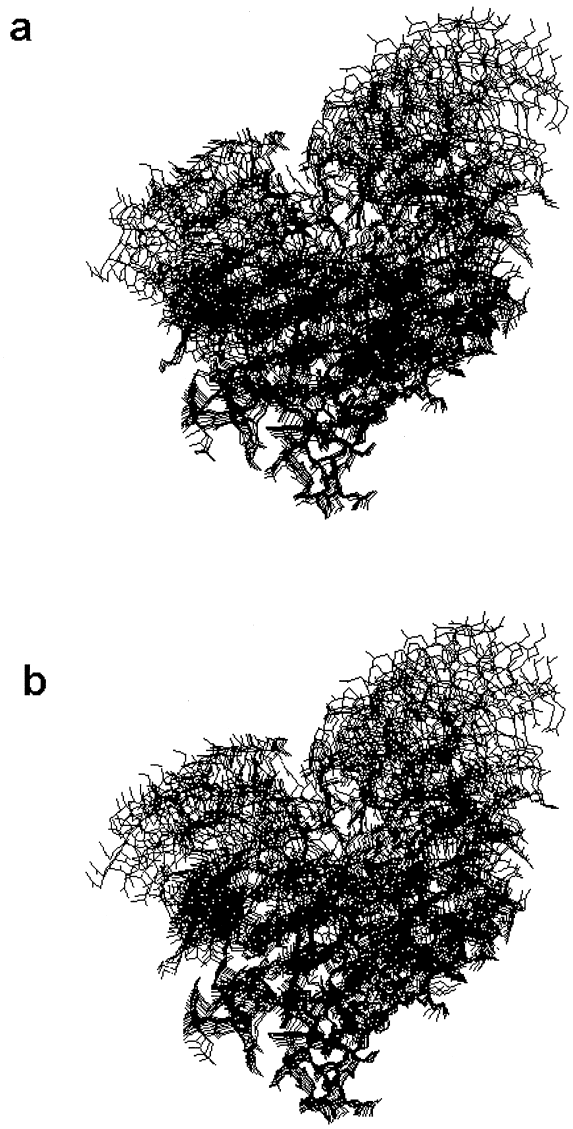

Figure 3. Unconstrained and alpha carbon pseudo bond constrained interpolation of adenylate kinase between the 'open' form (4AKE) and the 'closed' form (2ECK). Ten intermers are superimposed. In this example, the constraint has little effect. Contrast this with the interpolation of recoverin shown in Figure 4.

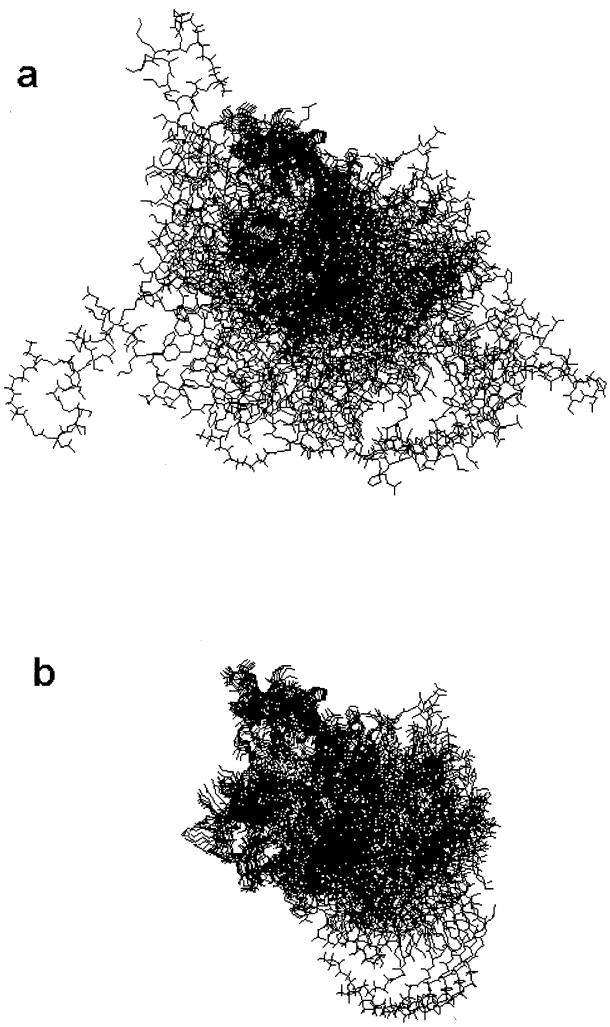

Figure 4. Unconstrained and alpha carbon pseudo bond constrained interpolation of recoverin between the calcium-free form (1IKU, model 1) and the calcium-bound form (1JSA, model 1). Without constraint, the N-terminal domain thrashes widely. With constraint, the extensive artifactual movement is abolished, the relative rotation of the domains is evident and the extraction of the N-terminal myristoyl group is smoothly modelled.

Interpol algorithm was modified so that during construction of any intermer, the positions of pseudo bonded atoms would be set before positioning the rest of the atoms. In effect, the positioning of an alpha carbon atom is pre-empted by pseudo bonding. The result is shown in Figure 1c. The movement of the C helix was considerably reduced. Figure 1d shows the effect on the interpolation of pseudo bonding the alpha carbon atoms of ALA71 to HIS72 and HIS50 to GLY51. The movement of the C helix was further reduced.

No other large backbone torsional differences between 1HHO and 1BZ0 are apparent, yet the movement of the A helix was unaffected by the introduction of the two pseudo bonds. This motion seems to be the result of a series of smaller torsional differences whose effects are amplified during assembly of the intermers. Having established that alpha carbon pseudo bonding can bypass these torsional differences, the interpolation was repeated with pseudo bonds between all of the alpha carbon atoms. This greatly reduced the artifactual movements, but not entirely. The interpolation was now influenced by differences in the torsion angles of the alpha carbon pseudo bonds. To counter this, higher-order pseudo bonding was introduced and best results were obtained when each alpha carbon atom (n) was pseudo bonded to the alpha carbon atom in the next residue (n + 1), the residue next but one (n + 2) and the residue next but 3 (n+4). The results of this three-layer alpha carbon pseudo bonding on the interpolation of 1HHO and 1BZ0 is shown in Figure 1e. All of the intermers had structures that were completely contained in the intervening space between the start and end structures and an animation based on this interpolation showed a smooth continuous movement with no reciprocation.

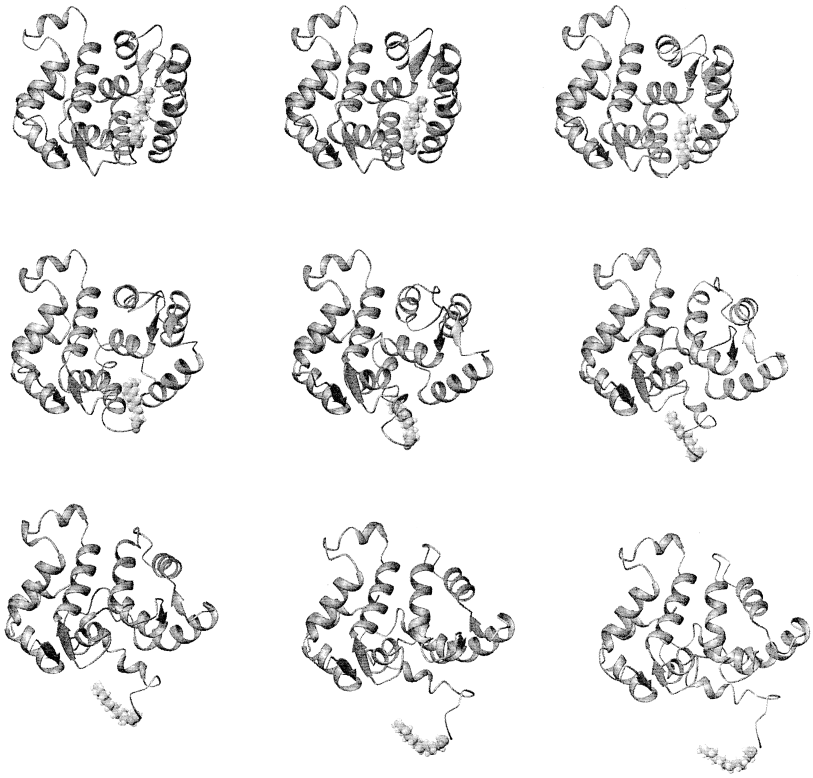

Figure 5. Nine frames from an AVI movie showing the extraction of the myristoyl group of recoverin. The interpolation was performed with the eight N-terminal residues excluded from constraint. All other residues were subject to restraint by alpha carbon pseudo bonding. Twenty intermers were produced and each was briefly energy minimized in Swiss-Pdbviewer (500 steps using conjugate gradients and the GROMOS96 force field). The PDB files were then visualized using the Windows version of Ribbons and the images produced were transferred to Paintshop Pro 5 to reduce the color depth to 8 bits per pixel. The AVI movie was then assembled in Microsoft VidEdit 1.1.

Java Interpol

The user interface of Java Interpol is shown in Figure 2. The user can select the two PDB files for interpolation and also the number of intermers to be produced. Alpha carbon pseudo bonding can be switched on or off. Another option is to pseudo bond the alpha carbon atoms in residues that are hydrogen bonded in both the start and the end structures. This option is useful for providing extra constraints for proteins that contain significant beta structure.

With alpha carbon pseudo bonding switched on, artifactual local deflections of the protein backbone can propagate no further than the next residue. However, in very flexible regions, this may result in unrealistic bond lengths and angles at some of the alpha carbon atoms. In these regions, an unconstrained interpolation might give better results. Atoms can be excluded from pseudo bonding by adding EXCLUD records to the PDB files. Each EXCLUD record can contain up to six ID numbers of excluded alpha carbon atoms. Similarly, sometimes it may be necessary to add extra constraining pseudo bonds. These can be inserted into the interpolation by adding PSEUDO records to the PDB files. The format of PSEUDO records is the same as that for CONECT records.

Interpolation of Other Proteins

The results of using Java Interpol for unconstrained and alpha carbon pseudo bonded interpolation for adenylate kinase and recoverin are shown in Figure 3 and Figure 4. These two proteins both display large movements. In the case of adenylate kinase, no attempt was made to model the sequential movements depicted by Vonrhein et al.[3] Rather, a simple interpolation was performed between the 'open' form (4AKE) and the 'closed' form (2ECK). The movement is a 'hinge' type movement with two wings folding in toward the center. Figure 3 shows that there was very little difference between the constrained and the unconstrained interpolations. Presumably, the movements do not involve large backbone torsional changes.

The case of recoverin is very different. This is a retinal calcium-binding protein that participates in the recovery phase of visual excitation and in adaptation to background light. The N-terminus is covalently blocked by a myristoyl group, which in the calcium-free state is sequestered in a deep hydrophobic pocket. When calcium binds to recoverin, the myristoyl group becomes extracted to the protein exterior, where it anchors the previously cytosolic protein to the membrane of the retinal disc.[11] The protein comprises four EF hands in two domains. The extraction of the myristoyl group involves a rotation of the N-terminal domain relative to the C-terminal domain.[12] Figure 4 shows the result of constrained and unconstrained interpolation between the calcium-free form of recoverin (1IKU, model 1) and the calcium-bound form (1JSA, model 1).

In the unconstrained interpolation, the C-terminal domain interpolates reasonably well, but the N-terminal domain shows massive artifactual movements, with the N-terminal section thrashing widely. The constrained interpolation abolished these large movements and an AVI movie produced from this interpolation clearly showed the extraction of the myristoyl group and the relative rotation of the two domains (Figure 5).

CONCLUSION

The distance between neighboring alpha carbon atoms varies remarkably little in most proteins due to the stiffness of the peptide plane. By introducing pseudo bonds between alpha

carbon atoms, variation in their distance can be constrained and high-energy distortions of the peptide unit can be prevented. Hence alpha carbon pseudo bonding provides a simple, but effective means of constraining interpolation in internal coordinate space. Within stable secondary structures, the distances between second- and third-nearest neighbor alpha carbon atoms vary little, and introducing higher-order pseudo bonding can be used to provide further restraint, although this should not be used in loop regions—hence the introduction of the EXCLUD record. This form of interpolation produces plausible (though not necessarily accurate) intermediate structures suitable for the production of animations and movies for teaching purposes. The intermediate structures produced by this Java Interpol will often require further refinement. At present this must be done by a second program, but further work is in progress to incorporate a crude geometric optimizer. Nonetheless, all of the steps from alignment of initial structures to the production of AVI movies can be carried out on personal computers readily available to teaching staff and educational developers.

ACKNOWLEDGMENTS

I am grateful to Jon Maber for providing the source code of his unpublished program and to Alan Berry, Mike Carson, Sandy Geddes, Robert Lancashire, and Eric Martz for critically examining some of the animations and movies produced during this study. The class files for Java Interpol and a selection of animations and movies can be accessed at http://www.bmb.leeds.ac.uk/agb/

REFERENCES

1 Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E. The Protein Data Bank. *Nucleic Acids Research* 2000, **28**, 235–242
2 Richardson, D.C., and Richardson, J.S. The kinemage: A tool for scientific communication. *Protein Sci.* 1992, **1**, 3–9
3 Vonrhein, C., Schlauderer, G.J., and Schulz, G.E. Movie of the structural changes during a catalytic cycle of nucleoside monophosphate kinases. *Structure* 1995, **3**, 483–490
4 Gerstein, M., and Krebs, W. A database of molecular motions. *Nucleic Acid Res.* 1998, **26**, 4280–4290
5 Kleywegt, G.J. Use of non-crystallographic symmetry in protein structure refinement. *Acta Cryst.* 1996, **D52**, 842–857
6 Olender, R., and Elber, R. Calculation of classical trajectories with a very large time step: Formalism and numerical examples. *J. Chem. Phys.* 1996, **105**, 9299–9315
7 Elber, R., Meller, J., and Olender, R. Stochastic path approach to compute atomically detailed trajectories: Application to the folding of C peptide. *J. Phys. Chem. B* 1999, **103**, 899–911
8 Fischer, S., and Karplus, M. Conjugate peak refinement: An algorithm for finding reaction paths and accurate transition states in systems with many degrees of freedom. *Chem. Phys. Lett.* 1992, **194**, 252–261
9 Guex, N., and Peitsch, M.C. SWISS-MODEL and the Swiss-PdbViewer: An environment for comparative protein modeling. *Electrophoresis* 1997, **18**, 2714–2723
10 Carson, M. Ribbons 2.0 *J.Appl.Cryst.* 1991, **24**, 958–961
11 Tanaka, T., Ames, J.B., Harvey, T.S., Stryer, L., and Ikura, M. Sequestration of the membrane-targeting myristoyl group of recoverin in the calcium-free state. *Nature* 1995, **376**, 444–447
12 Ames, J.B., Ishima, R., Tanaka, T., Gordon, J.I., Stryer, L., and Ikura, M. Molecular mechanics of calcium–myristoyl switches. *Nature* 1997, **389**, 198–202