A molecular viewer for the analysis of TLS rigid-body motion in macromolecules

Jay Painter and Ethan A. Merritt

A molecular viewer for the analysis of TLS rigid-body motion in macromolecules

Jay Painter and Ethan A. Merritt*

Biomolecular Structure Center, Department of Biochemistry, University of Washington, Seattle, WA 98195-7742, USA

Correspondence e-mail: merritt@u.washington.edu

TLS (translation/libration/screw) models describe rigid-body vibrational motions of arbitrary objects. A single-group TLS model can be used to approximate the vibration of an entire protein molecule within a crystal lattice. More complex TLS models are broadly applicable to describing inter-domain and other internal vibrational modes of proteins. Such models can be derived and refined from crystallographic data, but they can also be used to describe the vibrational modes observed through other physical techniques or derived from molecular dynamics. The use of TLS models for protein motion has been relatively limited, partly because the physical meaning of the refined TLS parameters is not intuitive. Here, a molecular viewer, *TLSView*, is introduced using OpenGL and based on the *mmLib* library for describing and manipulating macromolecular structural models. This visualization tool allows an intuitive understanding of the physical significance of TLS models derived from crystallographic or other data and may be used as an interactive tool to display and interpret inter-domain or other motions in protein structural models. *TLSView* may also be used to prepare, analyze and validate TLS models for crystallographic refinement.

1. Introduction

The use of TLS models in crystallography has arisen for two distinct reasons. In small-molecule crystallography, where it is typical to model the anisotropic vibrational displacement of each atom, TLS models are used to explain these individual atomic displacements in terms of concerted vibrations of a larger group of atoms. Thus, the individual ellipsoidal description of the atoms in a six-membered ring may be adequately explained by a rocking motion of the entire ring about some axis. In this context, the TLS model serves as a simplified description approximating a much more complicated underlying model that has already been refined against atomic resolution data. In protein crystallography such high-resolution data is rare and the lower resolution data which is obtained cannot support the complexity of a full model for anisotropic motion of each atom in the structure. In this context, the TLS description allows a much simpler model to be derived and refined directly against the lower resolution data, rather than being fitted *post hoc* to an already refined complex model. Even a very low resolution protein diffraction experiment yields sufficient observations to refine the 20 parameters that constitute a single-group TLS model. This is many orders of magnitude less than the six vibrational parameters per atom that would constitute a fully general anisotropic model.

research papers

The use of TLS models in protein crystallography is also attractive for a second reason, distinct from the limitations imposed by data resolution. Proteins are intrinsically flexible objects and even within a crystal lattice the surrounding volume of solvent permits many of the same large-scale vibrational modes and inter-domain flexibility that characterize the protein's behavior in solution. TLS models can describe these large-scale vibrational modes quite well and an optimal TLS description can yield considerable insight into the specific modes of conformational flexibility available to a particular protein. Single protein crystal structures, even at modest resolution, can thus suggest multiple modes of conformational flexibility. The coarsest such description, a single rigid group per molecule, is likely to be relevant only to the particular crystal lattice from which the structure was obtained. However, increasingly complex TLS models, consisting of multiple groups per molecule, can reveal biologically important modes of conformational flexibility.

Interpreting the physical significance of these models is currently a daunting task, since the information available to the structural biologist consists of a set of 20 numbers per assigned TLS group. A few existing programs, *e.g. TLSANL* (Collaborative Computational Project, Number 4, 1994; Howlin *et al.*, 1993), offer some assistance by generating static graphics that display the principal axes of the TLS components (see discussion below), but fall short of displaying intuitive interactive visualization of the implied motions. We introduce here a full three-dimensional molecular viewer, *TLSview*, that allows visualization and analysis of TLS models, offering a variety of animation modes and static display styles. The viewer is implemented on top of the freely available *Python Macromolecular Library* (*mmLib*; Painter & Merritt, 2004) and uses OpenGL to generate an interactive graphics display.

2. Methodology

Qualitative understanding of the TLS model and its interpretation as three translational displacements and three non-intersecting screw displacements is sufficient for understanding the visualizations available in *TLSView*. A full derivation of the TLS model parameterization is presented in detail by Schomaker & Trueblood (1968).

The atoms in a crystal are not perfectly arranged single points. In order to account for both the static disorder and dynamic vibration of atoms in the crystal, a crystallographic model must treat the position of each atom as a distribution of electron density about a mean position (x, y, z). The parameters describing this distribution are called atomic displacement parameters (ADPs; Trueblood *et al.*, 1996). Individual isotropic ADPs ('B factors') describe each atom as a spherical Gaussian distribution. Individual anisotropic ADPs (U^{ij}) describe the mean-square displacement of an atom about its mean position as a trivariate Gaussian distribution. This is the basis for the familiar 'thermal ellipsoids' representation used by programs such as *ORTEP* (Burnett & Johnson, 1996) and *rastep* (Merritt, 1999). If the displacement

of each atom is modeled independently, this requires one parameter (B) per atom for an isotropic model and six parameters (U^{ij}) per atom for an anisotropic model. A TLS rigid-body description is an alternative to treating each atom independently. It can be used to model the individual ADPs for each atom in the group by treating it as one point on a rigid body undergoing displacement.

In Euclidean space, a rigid-body displacement is fully described by a rotation matrix R and a translation vector t. The description can be simplified further using Chasles's Theorem to the form of a screw displacement; the rigid body is rotated about a specific vector while simultaneously being translated along that same vector by a magnitude proportional to the rotation angle (screw pitch). Given the location of a number of points on the rigid body before and after displacement, we can solve for R and t.

Reformulation of Euclidean rigid-body displacement to predict ADPs results in a probabilistic description of rigid-body motion requiring 20 unique parameters arranged into three 3×3 tensors: T, L and S. The translation tensor T is a symmetric 3×3 tensor analogous to the individual anisotropic ADP tensor U^{ij}. The T tensor is the anisotropic translational displacement common to all atoms in the rigid-body group. The libration (rotation) tensor L is also a symmetric 3×3 tensor with elements in units of radians^2, which are commonly converted to degrees^2 when reported. The libration tensor describes the rotational component of the rigid-body displacement. The T and L tensors are both trivariate Gaussian covariance tensors, implying that the matrix must be both symmetric and positive definite. Physical interpretation of TLS displacement requires that T and L be positive definite. If either matrix is not, then in some directions the tensor will predict a negative mean-square deviation (MSD), in violation of the mathematical assumptions of the TLS model. In particular, negative eigenvalues for L may arise from violation of the rigid-body assumption. This may be understood by considering the displacement of particles suspended in a elastic substance. When the elastic substance is rotationally flexed about an axis, the particles will be displaced in the plane perpendicular to the axis by the transfer of force through the elastic material. Particles closer to the axis will undergo larger displacements than particles further from the axis. Fitting TLS tensors to the displacement of the particles will result in an L tensor predicting a negative MSD rotational displacement in the direction of the axis coupled with a T tensor predicting a unreasonably large translational displacement in the plane perpendicular to the axis. The S tensor is not a Gaussian covariance tensor and is generally non-symmetric and difficult to interpret. It exists in the TLS model to account for a rotation–translation correlation which occurs when a rigid body undergoes rotation about three orthogonal axes that do not intersect at a common point.

The interpretation of a general TLS model as three translational displacements and three screw displacements is accomplished by a series of tensor transformations. Initially, the TLS parameters are calculated from an arbitrarily chosen origin. Once the parameters have been calculated, their origin

can be shifted to a fixed point calculated from the TLS parametric displacement called the center of reaction. This transformation minimizes the trace of T and ensures that the S tensor is symmetric. Minimization of the trace of T implies that a maximum amount of displacement is accounted for by the L and S tensors. The parameters from the L and S tensor are combined to calculate the screw displacements and the T tensor is reduced further to a new translation tensor T^r which is the absolute minimum translational component for the rigid body needed to account for the observed ADPs. This reduction thus maximizes the displacement contribution from the screw axes while minimizing the displacement contribution from pure translation.

The equations for calculating individual anisotropic ADPs from a TLS model require only the vector position of the atoms from the TLS origin as arguments and are linear with respect to the TLS parameters. Because of this, TLS parameter values are easily calculated for any given set of atoms with crystallographically refined ADPs using the method of least squares or weighted least squares. When there are more than 20 isotropic atoms or five anisotropic atoms the TLS parameters are over-determined and the predicted ADPs can be compared with the experimental ADPs. The predicted ADPs can also be partitioned into displacement contributed from group translation and displacement contributed from screw motion.

3. *TLSView* architecture

TLSView is distributed with the *Python Macromolecular Toolkit* (*mmLib*), a software package we have developed to manipulate macromolecular structure files and perform structural studies. The viewer has been tested on the Linu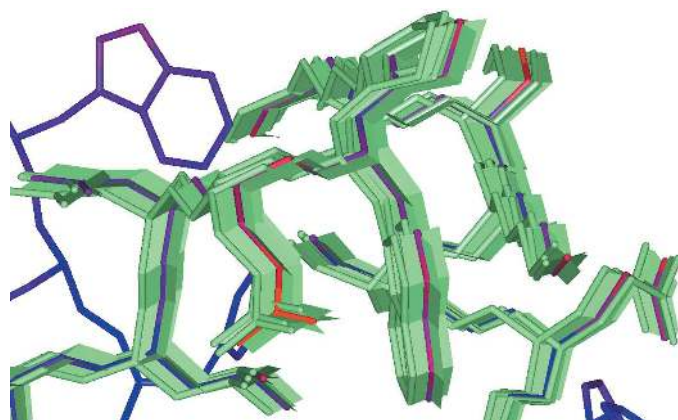x operating system, but should work on any Unix system with the appropriate system libraries and OpenGL hardware acceleration. The code will run on generic hardware, as long as the OpenGL primitives are supported. However, animation of multiple TLS groups using molecular-visualization styles with solids (spheres, rods) are computationally demanding and require a fast processor and an OpenGL hardware accelerated graphics card. The author uses a 3 GHz Pentium 4 with a GForce 5200 graphics card and finds this marginally adequate. The graphical user interface is written with the GTK+ toolkit (http://www.gtk.org/) using the PyGTK (http://www.pygtk.org) Python language bindings for the GUI library. The GtkGLExt and PyGtkGLExt (http://gtkglext.sourceforge.net/) GTK extension libraries are used to embed the OpenGL rendering window into the viewer. The OpenGL rendering system uses the PyOpenGL bindings (http://pyopengl.sourceforge.net/) and a Python extension module written in C to quickly tessellate solids.

3.1. Python TLS module

To support the development of *TLSView,* we wrote a GUI-independent Python library module to perform TLS-related tasks needed by the viewer. The module TLS.py contains classes to load and save TLS definition files, perform a post-refinement least-squares fit of atomic ADPs to TLS para-
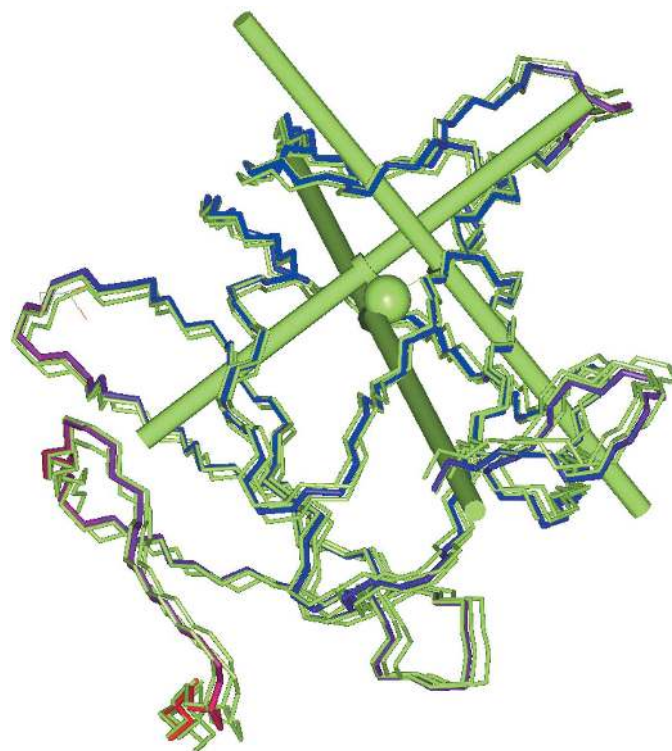

Figure 1
TLSView offers many visualization modes to represent group motion, some static and others using animation. Shown here is a composite of several snapshots from an animation. The base conformation of the molecule is shown as rods colored by temperature factor. Three green copies of the rod representation are captured from an animation of their motion along a trajectory consistent with the TLS screw displacements. A final superimposed snapshot shows a static rectangular surface for each bond, corresponding to the area swept out by a stick representation of that bond between the two extremes of the full trajectory.

Figure 2
The non-intersecting screw axes and T^r thermal ellipsoid for the single crystallographically unique chain of a dimeric protein (PDB code 1r75). Superimposed on this is a temperature-factor-colored backbone trace. Also shown are three green backbone traces displaced by the screw axes to a magnitude of 85% isoprobability. The length of the screw axes is drawn proportional to the rotational magnitude and the T^r ellipsoid is contoured at 85% isoprobability. The length of the thicker part of the screw axes is the displacement parallel to the screw axis as a result of rotation to a magnitude of 85% isoprobability.

meters, quantify the goodness of fit for TLS-predicted ADPs *versus* refined ADPs and all the tensor reductions needed to calculate the center of reaction and screw axes. The file input/output classes support *REFMAC*5 (Winn *et al.*, 2001) TLSIN/TLSOUT file format and the REMARK records found in PDB (Bernstein *et al.*, 1977) files of structures refined using *REFMAC*5's TLS refinement mode. Through this modular design, we have been able to provide well documented file format and computational classes for further applications. There are several example programs distributed with *mmLib* which perform complex TLS calculations independent of *TLSView*.

4. TLS visualization modes

TLSView was motivated primarily by the goal of providing a visualization tool that uses animation to display models of protein motion. However, the program also offers a wide variety of representations for the molecule itself and for the individual components of associated TLS models (Figs. 1, 2, 3 and 4). The anisotropic atomic ADPs calculated from a TLS model may be visualized using any of the traditional visualizations such as thermal axes, thermal ellipsoids (Burnett & Johnson, 1996) and peanuts (Hummel & Hauser, 1990). A given atom's predicted ADP tensor is the sum of translational and screw displacement from the TLS model. In *TLSView* these contributions are drawn separately so that they can be compared. When using the thermal ellipsoid or peanut visualization, the smaller component of the ADP is usually hidden by the larger. Transparency settings for thermal ellipsoid and peanut visualizations are provided so that the smaller component of the ADP may be seen. The screw displacement axes may be visualized as in the *CCP*4 program *TLSANL*, which represents the screw axes as rods with a length proportional to their mean-square deviation rotational magnitude. *TLSView* supports these visualizations with greater detail than provided by previous programs and introduces several new visualization modes useful for multiple TLS groups defined as segments of a protein chain.

4.1. Default visualization

The default *TLSView* visualization of a TLS group is drawn from the center of reaction. If the TLS parameters loaded from a TLSOUT or PDB file are not already shifted to their center of reaction, *TLSView* shifts them automatically. The reduced translation tensor T^r is drawn as a 50% isoprobability thermal ellipsoid at the center of reaction and the three screw displacement axes are drawn with a length proportional to their mean-square deviation rotational magnitude. When multiple TLS groups are defined within a protein chain, each is automatically assigned a unique color to be used by all its TLS visualizations. To help identify the residues belonging to each TLS group, a backbone trace is drawn over its residues. Fig. 2 shows the default TLS visualization applied to one monomer of the dimeric protein structure (PDB code 1r75) using a single TLS model for the entire monomer. In this figure, three copies of the TLS colored backbone traces are shown as displaced independently by the screw axes. A temperature-factor-colored backbone trace is drawn at the actual structure coordinates. The TLS backbone trace is drawn by a subclass of *mmLib*'s GLAtomList renderer. This rendering object draws all of the standard structural visualization styles and the user can choose to visualize the TLS group atoms with them. It also implements a new visualization called COR-Fan which draws a series of triangles along the backbone atoms using the center of reaction as a common vertex. This is useful when there are a large number of TLS groups defined along a protein chain. Fig. 4(*b*) shows a COR-Fan visualization of HIV protease in which the screw axes and T^r thermal ellipsoid are hidden for clarity.
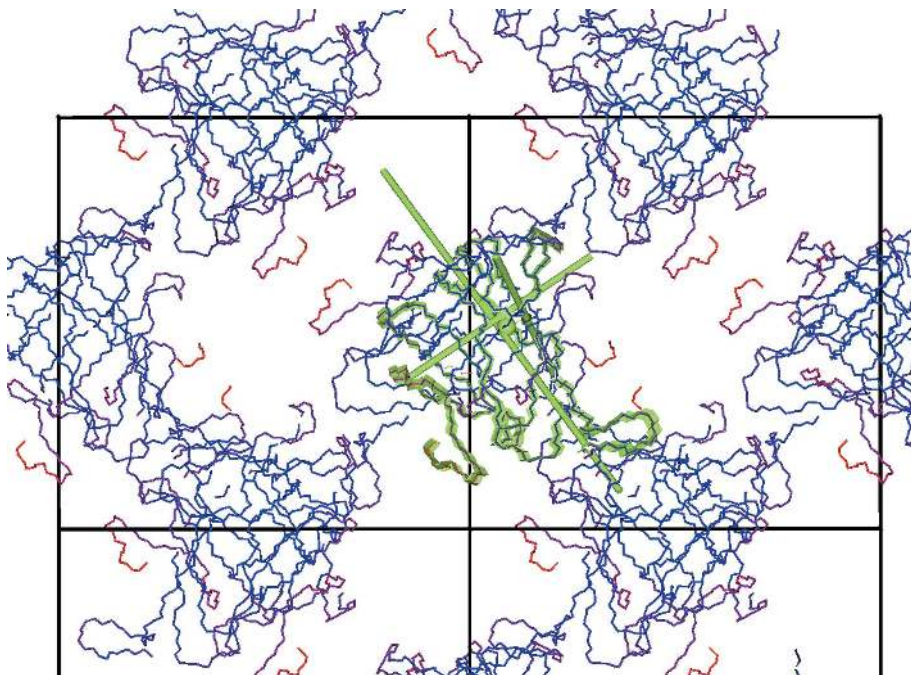

Figure 3
The same view and TLS model for 1r75 as in Fig. 2 is shown here in the context of the crystal lattice. During an interactive session, the relationship of bulk TLS motion to lattice packing is seen most easily using animation, but here we show a static representation. Symmetry-related molecules are represented as a backbone trace colored by temperature factor. The central molecule (green) is superimposed with the TLS non-intersecting axes and three screw displacement surfaces at 85% isoprobability along the backbone. Each segment of the green surface represents the trajectory the backbone would take if displaced by its corresponding screw axis to the 80% isoprobability level. All elements of the view can be animated.

4.2. Animation of screw motion

The three TLS non-intersecting orthogonal screw axes are drawn as rods with a length proportional to their

rotational mean-square deviation. The screw pitch for each axis is visualized by a segment of larger radius at the center of each axis. The length of the segment is equal to the translational displacement which occurs when the screw axis is rotated to the current isoprobability setting. Although knowing the location and magnitude of the screw axes is useful, it is more interesting to view the effect of the screw displacements on the atomic model. To accomplish this, we have developed two visualizations for the screw displacements. The first visualization draws a surface tracing the trajectory of the model atoms as they are displaced about a screw axis. This is performed by incrementally applying the screw displacement to the TLS group atoms and drawing a quadrilateral between the previous and current increments of
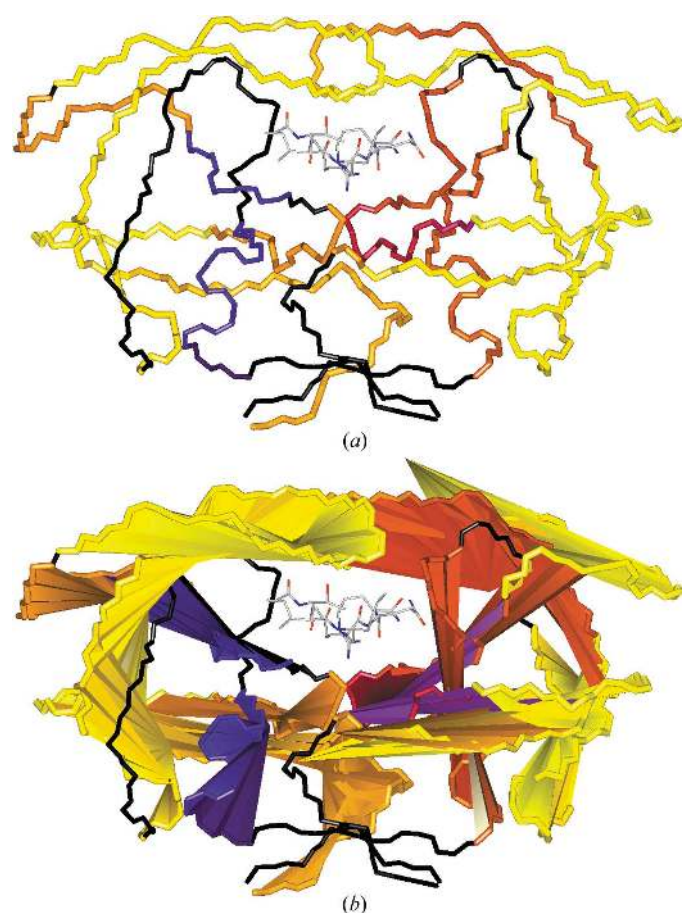
all bonded atoms. The surface is drawn up the a given isoprobability level of screw displacement and then repeated for the other two screw axes. Since these surfaces are formed by the principal screw displacements, their interpretation is equivalent to a visualization of translational displacements along principal axes. The second visualization performs the same incremental displacement used to draw the screw displacement surface, but uses it to interactively displace copies of the atomic model simulating the vibrational motion described by the screw axes. A sine function is used to rock the atomic model during animation, but is used only for convenience and in no way implies classical harmonic motion.

5. Generation of TLS parameters from refined *B* values

5.1. Fitting TLS parameters to groups of atoms previously refined with isotropic or anisotropic ADPs

TLSView provides a convenient interface to *mmLib* routines that generate TLS parameters by a least-squares fit directly from crystallographic models containing either isotropic or anisotropic thermal parameters for individual atoms. This can be particularly useful to generate an initial TLS model for refinement of protein structures at medium to low resolution starting from individual or grouped isotropic *B* values. Refinement that starts from such a pre-fit TLS model is more stable numerically than refinement that evolves the TLS parameters from a starting state with all TLS parameters set to zero and thus may permit successful TLS refinement in cases that are otherwise poorly behaved.

TLSView can also be used to explore the generation of more complex vibrational models involving multiple TLS groups. Here again this may be useful as a starting point for continued crystallographic refinement or as a guide to the interpretation of a previously refined structure. Such *post hoc* fitting of multiple TLS groups is commonly used to model vibrational modes in small-molecule crystallography (Schomaker & Trueblood, 1998), but can also be used to explore possible large-scale flexibility in proteins (Wilson & Brunger, 2000; Chaudhry *et al.*, 2004; Ming *et al.*, 2002; Papiz *et al.*, 2003; Bernett *et al.*, 2004).

5.2. Identification of regions not adequately described by TLS

TLSView can assist in identifying regions within a protein structure whose vibrational modes can be described by separate TLS groups. A first step in this process is the use of an algorithm, *TLS Running Segment Fit* (*TLS-RSF*), that rapidly identifies those portions of the structure whose ADPs are well described by TLS motion and those portions whose ADPs do not conform to a TLS model. This algorithm works by fitting TLS parameters to successive overlapping segments of *w* contiguous residues spanning the entire length of the protein chain. Segment *i* therefore consists of residues *i* to $(i + w - 1)$. The TLS-predicted atomic ADPs for each segment are then compared with the refined ADPs from the input model. This approach can be applied either to high-resolution models

Figure 4
Visualizations output from the *TLS-RSF* algorithm applied to the 2.3 Å resolution structure of an HIV protease–inhibitor complex (PDB code 4hvp) using a segment width of eight residues. (*a*) Each residue of the backbone trace is colored according to the best goodness of fit among the eight overlapping segments to which it belongs. The color gradient runs from dark blue (poorest fit to a TLS model) through yellow (best fit to a TLS model). Long runs of yellow indicate portions of the chain whose displacements are well described by TLS as rigid-body motions. However, these continuous regions of good TLS fit may span multiple TLS groups with very different implied motions. (*b*) The COR-Fan visualization mode draws triangles connecting successive backbone atoms to the center of reaction of the TLS group fitted to each segment to which the atoms belong. The fans help to identify runs of residues whose optimal TLS descriptions have similar centers of reaction. Such a run appears in the visualization as a multi-segmented fan emanating from a single cluster of origins, indicating that the residues in this run act as a group.

refined with anisotropic ADPs or to models at any resolution refined with isotropic ADPs.

The unconstrained least-squares algorithm used by *TLS-RSF* may yield negative eigenvalues for the T and L tensors of some segments. Such tensors suggest a violation of the rigid-body assumption even if the resulting fit of predicted to observed ADPs for these groups is good. For this reason, these segments are automatically discarded and the remaining valid segments are visualized and colored by their goodness of fit. The default segment width w is six residues, but can be set by the user. The algorithm can optionally omit side-chain atoms and atoms with partial occupancy from the TLS least-squares

residual for each segment. Fig. 4 shows the initial visualization output by the *TLS-RSF* algorithm applied to a 2.3 Å resolution structure of HIV protease (PDB code 4hvp) refined with isotropic thermal parameters and no treatment of TLS (Miller *et al.*, 1989). The well known motility of the 'flap' regions is evident as being suited to description as separate TLS groups; note in particular the yellow fan in the foreground at the upper left of Fig. 4(*b*).

5.3. Partitioning a structure into multiple TLS groups

Having first identified regions which are not suitable candidates for treatment as part of a larger TLS group, one may proceed to partition the remainder of the structure into some number of non-overlapping TLS groups. There are several approaches to the identification of protein domains or smaller regions which act as a single concerted group with respect to a major vibrational mode. Simple visual assignment of likely domain boundaries may be possible, particularly if multiple conformations of a possible hinge motion are captured in multiple crystal structures (Sali *et al.*, 1992). If the data is of sufficiently high resolution to support refinement of
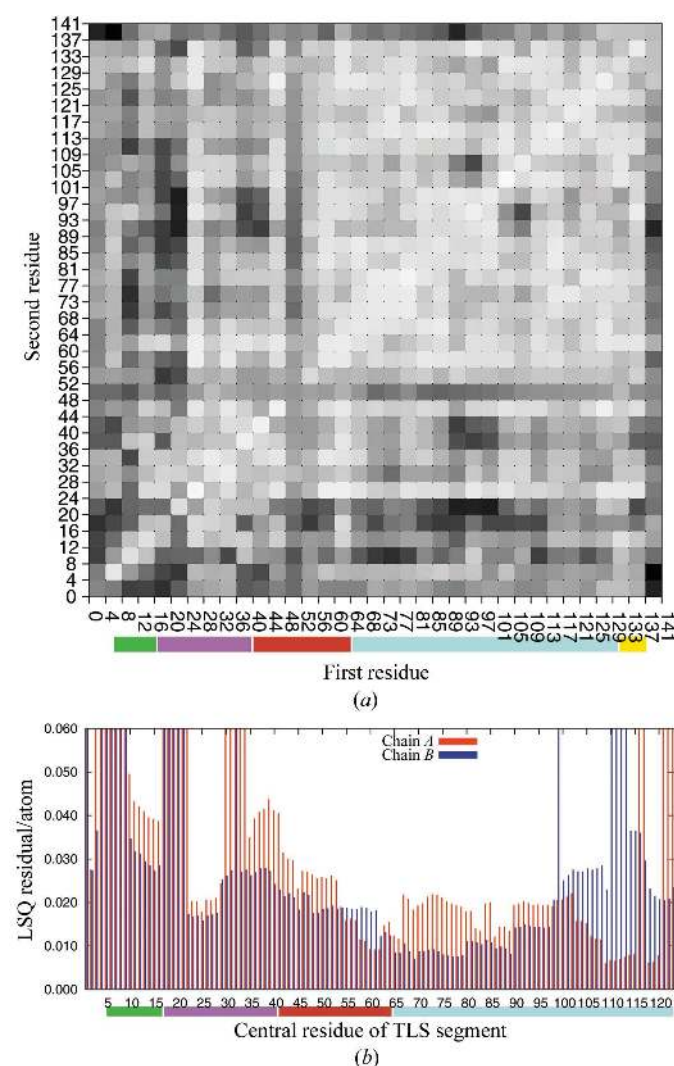

Figure 5
Partitioning the 1.1 Å resolution structure of human FGF-1 (PDB code 1rg8) into multiple TLS groups. (*a*) Output from *ANISOANL* applied to the backbone atoms of chain *A* using four-residue segments. This may be compared with Fig. 5 of Bernett and coworkers, which achieves better results through custom modification of *ANSOANL*'s algorithm (Bernett *et al.*, 2004). (*b*) Output from the *TLS-RSF* algorithm applied to both chains *A* and *B* using a segment width of 20 residues. The residue number along the *x* axis corresponds to the central residue of the segment. Regions that are well described by a one or more 20-residue TLS groups (residual less than some threshold value) may be subjected to further analysis in order to determine an optimal partition into a small number of TLS groups, with each TLS group comprising as large a region as possible. (*c*) Visualization of the screw displacement surfaces from an optimal partition into six TLS groups per chain as reported by the external automated tool *TLSMD* (unpublished work). Only chain *A* is shown. Each run of residues corresponding to a separate TLS group is shown in a distinct color and these same colors have been used *post hoc* to annotate the output shown in (*b*). The TLS model fit to residues at the N-terminus has a distinctly worse residual than the others, which bears out the generally higher residuals seen for the N-terminus of both chains in (*b*). Because these residues are poorly fit by a TLS model, only the backbone trace (gray) is shown. Notice that the T^r ellipsoids drawn for the red and cyan groups, corresponding to residues 41–64 and 65–131, lie at almost the same point in space. This is an indication that these two groups might be adequately described by a single group spanning this entire residue range.

a individual anisotropic ADPs for each atom, then the atoms moving as a single group can in theory be identified by pair-wise comparison. Two atoms belonging to the same rigid group should have identical displacements in the direction of the vector between them (Rosenfield *et al.*, 1978). This principle forms the basis for programs such as *THMA* (Schomaker & Trueblood, 1998) and *ANISOANL* (Winn, 2001). *THMA* was designed only for small-molecule work, but *ANISOANL* is designed for routine application to any protein structure refined with anisotropic ADPs. Unfortunately, the matrix of pairwise displacement magnitudes generated from a large protein structure is usually difficult to interpret. A particularly favorable case was reported for human FGF-1 by Bernett and coworkers, who proceeded to validate potential assignments of individual β-strands to larger TLS groups by comparing the ADPs predicted by such a TLS model with the observed ADPs from refinement at 1.1 Å resolution (Bernett *et al.*, 2004). The output of *ANISOANL* applied to chain *A* of the FGF-1 structure is shown in Fig. 5(*a*). The output of *TLSView*'s *TLS-RSF* analysis for the same protein is shown in Fig. 5(*b*).

The COR-Fan visualization mode of *TLSView* may directly suggest runs of residues that are jointly well described by a common TLS group (Figs. 4*b* and 5*c*). The fan visualization can also be used to quickly evaluate or adjust manual assignment of residues to a common TLS group made on the basis of an external program such as *ANISOANL*. Finally, *TLSView* can be used in conjunction with a separate automated tool *TLSMD* for choosing TLS groups based on a the individual ADPs from a partially or fully refined structural model (Fig. 5*c*). Generation of optimal models containing multiple TLS groups is a complex topic, which we will describe along with the *TLSMD* program in a separate publication.

6. Summary

TLSView is a user-friendly application for viewing macro-molecular structures with TLS rigid-body displacement models. It supports static visualizations of TLS-calculated thermal ellipsoids and screw axes found in previous programs and extends this by providing generation of screw-trajectory surfaces and interactive animation of the structural model. These TLS visualizations emphasize the separation of trans-lational displacement from screw displacement, which may provide valuable information supporting a hypothesis of shear or hinge conformational changes in domain or loop regions of a protein. The potential utility of TLS models in quantifying such conformational changes is hampered by the need to correctly partition the protein chain into multiple rigid groups, each with a corresponding set of TLS parameters. The *TLS-RSF* algorithm described here is useful for the qualitative identification of such groups, but we will report elsewhere

more sophisticated techniques for globally optimal selection of TLS groups. *TLSView* may also be used to construct, analyze or validate TLS models to be used in macromolecular refinement.

7. Availability

TLSView is being developed as part of the *mmLib* library (Painter & Merritt, 2004), a collaborative open-source project that is currently hosted on SourceForge (http://pymmlib.sourceforge.net/). The source code is currently released under the Artistic License, but it is our intention to be as flexible as possible on licensing issues. Installation of the library and viewer is straightforward, but requires that the user's machine support Python and OpenGL.

References

Bernett, M. J., Somasundaram, T. & Blaber, M. (2004). *Proteins Struct. Funct. Genet.* **57**, 626–634.

Bernstein, F., Koetzle, T., Williams, G., Meyer, E., Brice, M., Rodgers, J., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J. Mol. Biol.* **112**, 535–542.

Burnett, M. N. & Johnson, C. K. (1996). *ORTEP-III. Oak Ridge Thermal Ellipsoid Plot Program For Crystal Structure Illustrations.* Oak Ridge National Laboratory Report ORNL-6895. Oak Ridge National Laboratory.

Chaudhry, C., Horwich, A. L., Brünger, A. T. & Adams, P. D. (2004). *J. Mol. Biol.* **342**, 229–245.

Collaborative Computational Project, Number 4 (1994). *Acta Cryst.* D**50**, 760–763.

Howlin, H., Butler, S. A., Moss, D. S., Harris, G. W. & Driessen, H. P. C. (1993). *J. Appl. Cryst.* **26**, 622–624.

Hummel, W. & Hauser, J. (1990). *J. Mol. Graph.* **8**, 214–229.

Merritt, E. A. (1999). *Acta Cryst.* D**55**, 1109–1117.

Miller, M., Schneider, J., Sathyanarayana, B. K., Toth, M. V., Marshall, G. R., Clawson, L., Selk, L., Kent, S. B. & Wlodawer, A. (1989). *Science*, **246**, 1149–1152.

Ming, D., Kong, Y., Lambert, M. A., Huang, Z. & Ma, J. (2002). *Proc. Natl Acad. Sci. USA*, **99**, 8620–8625.

Painter, J. & Merritt, E. (2004). *J. Appl. Cryst.* **37**, 174–178.

Papiz, M. Z., Prince, S. M., Howard, T., Cogdell, R. J. & Isaacs, N. W. (2003). *J. Mol. Biol.* **326**, 1523–1538.

Rosenfield, R. E., Trueblood, K. N. & Dunitz, J. D. (1978). *Acta Cryst.* A**34**, 828–829.

Sali, A., Veerpandian, B., Cooper, J. B., Moss, D. S., Hoffmann, T. & Blundell, T. L. (1992). *Proteins Struct. Funct. Genet.* **12**, 158–170.

Schomaker, V. & Trueblood, K. N. (1968). *Acta Cryst.* B**24**, 63–76.

Schomaker, V. & Trueblood, K. N. (1998). *Acta Cryst.* B**54**, 507–514.

Trueblood, K. N., Bürgi, H.-B., Burzlaff, H., Dunitz, J. D., Gramaccioli, C. M., Shmueli, U. & Abrahams, S. C. (1996). *Acta Cryst.* A**52**, 770–781.

Wilson, M. A. & Brunger, A. T. (2000). *J. Mol. Biol.* **301**, 1237–1256.

Winn, M. (2001). *CCP4 Newsl. Protein Crystallogr.* **39**.

Winn, M. D., Isupov, M. N. & Murshudov, G. N. (2001). *Acta Cryst.* D**57**, 122–133.