DINAMO: A COUPLED SEQUENCE ALIGNMENT EDITOR/MOLECULAR GRAPHICS TOOL FOR INTERACTIVE HOMOLOGY MODELING OF PROTEINS

MARC HANSEN[†‡], JESSE BENTZ[*‡],
ALBION BAUCOM[†], LYDIA GREGORET[§]
Departments of Biology[*], Chemistry & Biochemistry[§], and Computer Science[†]
University of California, Santa Cruz, CA 95064
http://tito.ucsc.edu/

Gaining functional information about a novel protein is a universal problem in biomedical research. With the explosive growth of the protein sequence and structural databases, it is becoming increasingly common for researchers to attempt to build a three-dimensional model of their protein of interest in order to gain information about its structure and interactions with other molecules. The two most reliable methods for predicting the structure of a protein are homology modeling, in which the novel sequence is modeled on the known three-dimensional structure of a related protein, and fold recognition (threading), where the sequence is scored against a library of fold models, and the highest scoring model is selected. The sequence alignment to a known structure can be ambiguous, and human intervention is often required to optimize the model. We describe an interactive model building and assessment tool in which a sequence alignment editor is dynamically coupled to a molecular graphics display. By means of a set of assessment tools, the user may optimize his or her alignment to satisfy the known heuristics of protein structure. Adjustments to the sequence alignment made by the user are reflected in the displayed model by color and other visual cues. For instance, residues are colored by hydrophobicity in both the three-dimensional model and in the sequence alignment. This aids the user in identifying undesirable buried polar residues. Several different evaluation metrics may be selected including residue conservation, residue properties, and visualization of predicted secondary structure. These characteristics may be mapped to the model both singly and in combination. DINAMO is a Java-based tool that may be run either over the web or installed locally. Its modular architecture also allows Java-literate users to add plug-ins of their own design.

---

‡ These authors contributed equally to this work.

1. Introduction

1.1 The need for a modeling tool

A three-dimensional model of a protein can be invaluable in guiding experiments to investigate the function of the protein at the level of individual amino acid residues. Building such a model, however, is not straightforward. Despite the fact that nearly all bench biologists are facile with personal computers, most have not had extensive experience with three-dimensional protein modeling. To build a homology model of a protein based on its sequence similarity to a protein whose structure has been solved experimentally, a researcher may have to consult a colleague with a molecular graphics workstation and software. Amino acid substitutions may have to be made manually by comparing a printed sequence alignment to the known structure. In short, model building is far from automatic and depends not only on the researcher's perseverance, but also on the availability of software and on the cooperation of others.†

Furthermore, recent developments in the area of protein fold recognition are still unknown to most molecular biologists. A powerful method of protein structure prediction called "threading" is based on the idea that there exist a limited number of protein folds, and that the fold of a given protein will resemble the fold of a protein whose structure has been solved despite very little or no detectable sequence similarity[1]. Threading methods are typically able to choose the correct fold from a library of representative protein folds about fifty percent of the time. The alignment to the correct fold, however, often has errors and requires hand-correction.

We have developed an interactive protein model building tool called DINAMO (pronounced De-nah-mo). This tool is intended for use by a broad spectrum of researchers, including biologists as well as computer scientists, who also may have little experience with three-dimensional molecular graphics. DINAMO includes a Java-based sequence alignment editor which is coupled to a graphical display. Sequence alignments may be input in the standard FASTA format. DINAMO also accepts homology and threading-based alignments from PHD[2, 3]. The alignment is dynamic, in that as the user makes modifications, the effect of these changes is reflected in both the editor and the three-dimensional model.

---

† To appreciate the complexity of homology modeling using the UCSF MidasPlus modeling package, visit `http://www.cgl.ucsf.edu/midas-info/faq.html`

1.2 Similar applications

The concept of coupling sequence alignment to three-dimensional molecular graphics has been applied previously. Two commercial products currently available are LOOK from Molecular Applications Group (Palo Alto, CA) and HOMOLOGY from Molecular Simulations Inc. (San Diego, CA). LOOK is a stand-alone molecular modeling package and HOMOLOGY is a module for MSI's molecular graphics program Insight II. These programs incorporate the ability to generate detailed, atomic resolution models of the target sequences and include side chain rotamer optimization, loop building, and/or energy refinement. DINAMO differs from these products in three key ways. First, low-resolution models are generated. Since knowledge of the spatial proximity of amino acid side chains is usually sufficient to guide experiments, and since detailed side chain orientation is difficult to predict, especially at the surfaces of proteins[4] we chose to omit this added complexity from DINAMO. We also do not attempt to model unconserved loops but do note their location with visual cues. Second, there are no automatic refinement calculations performed by DINAMO either at the level of the sequence alignment or at the level of the three-dimensional structure. The user makes the final decision about how to align sequences based on the results of various assessment tools and his or her scientific reasoning. If the sequences of interest are unknown or weakly homologous, it is also possible to use Rost's and Sander's PHD program. Third, DINAMO is written in Java and is therefore highly portable. It can be run either as a stand alone application or as an applet via the web. It does not require the acquisition of costly supporting software or difficult installation procedures. For the web version of DINAMO, the only accompanying software required is the plug-in Chime[5, 6], the installation of which is automated by the latest web browsers. For heavy use, the stand alone version of DINAMO takes advantage of Java capabilities not available to applets such as integrated native C code for speed, and file I/O for loading and saving alignments.

Two freely-available applications similar to DINAMO are CINEMA, an interactive multiple sequence alignment editor with a newly-incorporated three-dimensional display and ANALYST [7], a tool for evaluating threading alignments produced by THREADER[8]. DINAMO differs from CINEMA in that the sequence alignment editor and the molecular graphics display are more closely linked: DINAMO allows the user to visualize the results from multiple alignment analysis algorithms both in the sequence editor and in the molecular graphics display. Another difference between DINAMO and CINEMA is that there is no code dependent on the web's Common Gateway Interface (CGI) protocol, in order to facilitate local installation and use. ANALYST is entirely an evaluation tool that has no editing capability, though alternative threadings are compared using visual tools like DINAMO's.

2. Overview of Architecture

DINAMO consists of four major components: an alignment editor, a display mapper, assessment tools ("plug-ins"), and a molecular graphics display (Figure 1). The user interacts directly with three of these components: the sequence alignment editor, the display mapper, and the molecular graphics display. He or she can adjust the alignment in the alignment editor, use the display mapper to select which assessment plug-ins to use and determine how their results are shown, and examine the results of the alignment modifications by manipulating the model in the molecular graphics window.

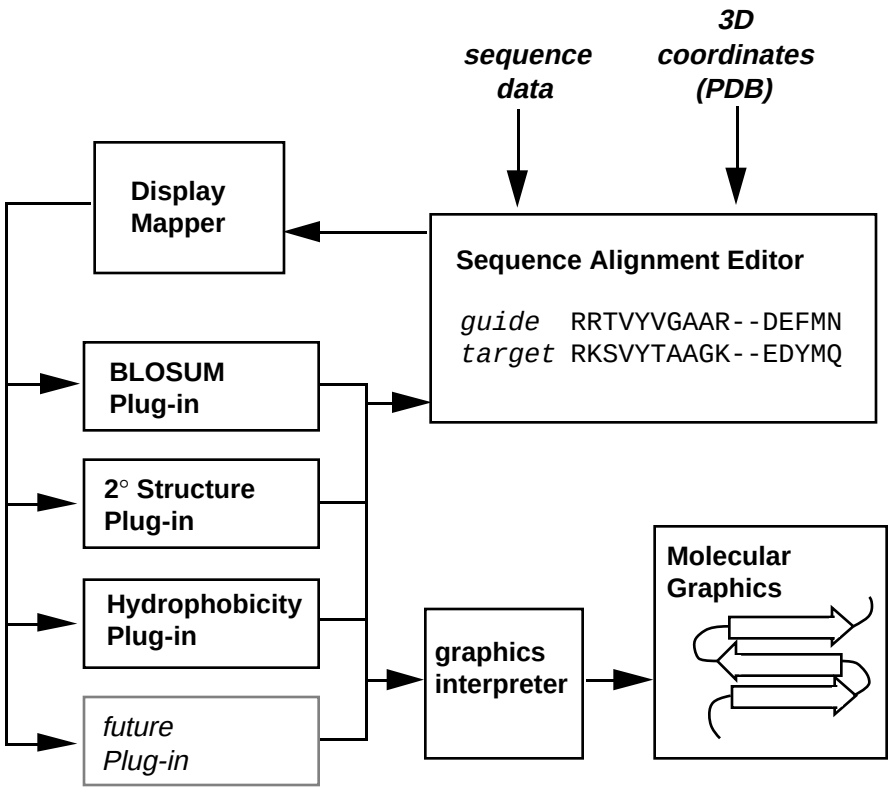

Figure 1. Information flow between DINAMO components.

Behind the scenes, the sequence alignment editor sends alignment information to the display mapper each time a change is made to the alignment. The mapper, in turn, relays this information to the active plug-ins. The plug-ins return assessment results back to the alignment editor and a graphics interpreter that communicates with the molecular graphics display. The purpose of the graphics interpreter is to allow future incorporation of different molecular graphics packages.

The DINAMO user interface is shown in Figure 2 as launched from Netscape Navigator 3.01 on a Silicon Graphics Indy running Irix 6.2. The three-dimensional image of the molecule being modeled is shown in the main window. The sequence alignment editor and display mapper are in separate windows. Each of these windows is launched from a central Java applet.

3. Sequence alignment editor

The sequence alignment editor forms the basis of the DINAMO package. It may be run in a stand-alone mode or used with the graphics package and/or one of several different assessment plug-ins. In its simplest form, the editor allows the researcher to optimize an alignment by adjusting the position of residues in a novel sequence relative to a guide, or template sequence (i.e. the sequence of the experimentally-determined structure). If three-dimensional structural information about the guide model is desired, it is computed at the beginning of a session directly from the structure's Protein Data Bank (PDB) coordinates[9, 10].

At the beginning of a modeling session, the user pastes a multiple sequence alignment into an input window. Sequences must be in FASTA format (readily available from protein sequence databases) but may be either unaligned or prealigned. In addition, alignments produced by PHD[3] may also be used.

The first sequence in the alignment is considered the "guide" sequence against which other sequences are compared. In general, this guide sequence has a known structure. The PDB coordinates are retrieved by DINAMO and the model displayed in the molecular graphics window. However, this is not mandatory: DINAMO can be used as a stand alone multiple sequence alignment editor, though alignment evaluation is relatively limited without the three-dimensional structure.

Each sequence in the alignment is displayed as a series of residue tiles. Each tile has its own set of properties which reflect information returned by plug-ins about the residue in the context of the sequence (or model, when activated). Currently, such information can be mapped to a residue tile by background color, shape, and text color.

The editor implements a set of radio buttons which allow users to select the sequence to be analyzed and displayed by the graphics module. When a model button is activated, the editor sets a variable in the alignment object indicating

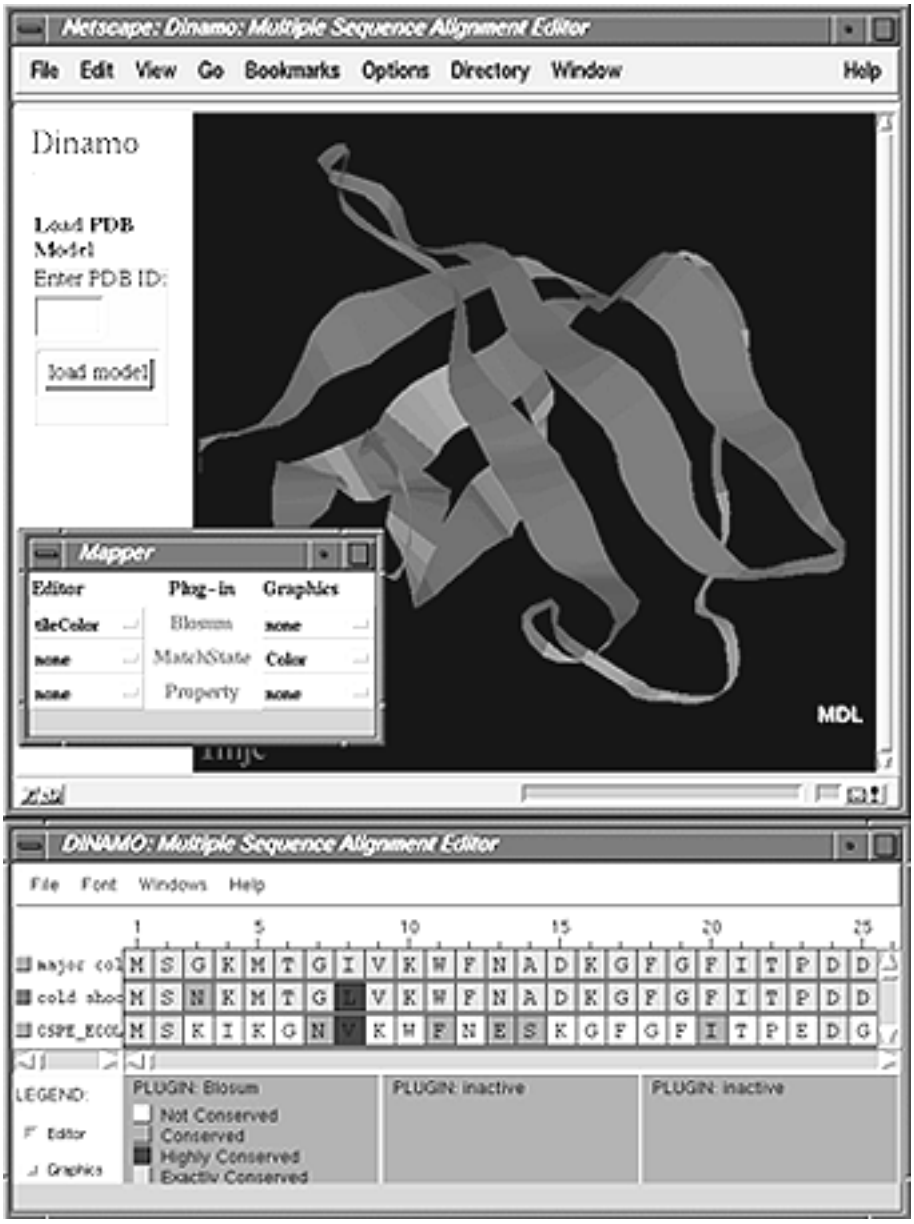

Figure 2. The DINAMO user interface.

which sequence was selected, and sends the alignment to be analyzed by the graphics plug-ins.

The alignment information is stored in the form of an alignment object. Alignment objects contain an array of strings representing each sequence in the alignment. In addition to the sequences themselves, the sequence names and the PDB identifier for the guide sequence are also stored. Information regarding the last sequence modified or selected for display in the graphics module is also contained in the alignment object.

Each time the alignment is edited, the editor updates a field in the alignment object to reflect which sequence was edited. After setting the appropriate field values in the alignment object, the editor sends the display mapper a reference to the modified alignment. This allows mapped properties to be dynamically updated as the alignment is adjusted. For example, if the user activates the residue conservation plug-in and chooses to map that attribute to color, each residue tile will be colored according to its degree of conservation to the corresponding residue in the guide sequence. As the user adjusts the alignment, the color of each tile in the guide and target sequences are dynamically updated to reflect the conservation of those residues.

The web version of DINAMO allows alignments to be mailed to the user from the web server if it implements the Simple Mail Transfer Protocol (SMTP). The stand alone version allows alignments to be both read from and written to the local file system.

4. Display Mapper

The display mapper (Figure 2.) is launched from the sequence alignment window. The mapper presents a menu which shows available plug-ins. It allows the user to activate or deactivate the plug-ins and select how the information returned by each plug-in is mapped to display options in the editor and graphics modules. Upon startup, only the plug-in which colors residue tiles according to residue properties (charge, polarity, etc.) is active. The user may select other or additional evaluation metrics and thus control both how much information is displayed and how that information is visualized.

The display mapper accomplishes this tracking via the use of legend objects associated with each plug-in. Every plug-in creates its own legend object and initializes it with its default display values (usually colors). This legend object saves both the editor and graphics display options for that plug-in only. If these display options change, the display mapper updates the corresponding plug-in's legend.

In addition to legend objects, the display mapper package also controls an alignment router. The alignment router takes an alignment sent from the editor and relays it to the active plug-ins based on information contained in a set of hash tables maintained within the display mapper.

Along with the alignment, the display mapper sends the plug-ins a reference to the client requesting the results. The determination as to whether the plug-ins will need to send their results to the editor, the graphics display, or both is based on the information received from the editor and the status of the plug-ins' legends.

5. Plug-ins

Plug-ins extend the capability of the sequence alignment editor by providing assessment criteria. Each plug-in has a unique algorithm for assessing alignment quality. Several plug-ins may be used and displayed simultaneously by mapping the results of a particular plug-in to a unique attribute in both the editor and molecular graphics packages.

Some examples of algorithms which we have implemented as plug-ins include: coloring both the three-dimensional structure and alignment by residue conservation (using the BLOSUM 62 matrix[11]), displaying secondary structure in the alignment editor, and highlighting residues by hydrophobicity.

The plug-ins component of DINAMO has been separated from the rest of the application in order to facilitate the creation and use of plug-ins with new functionality. To create a new plug-in, the programmer does not need to recompile any of the existing DINAMO code. The programmer must add the new plug-in to the dinamo.plugins package. This is accomplished by simply following the DINAMO plug-in Application Programmer Interface (API) and saving the code for the plug-in in the dinamo/plugins directory. Dinamo takes advantage of Java's ability to create instances of objects given the name of the object as a string. In the standalone version, DINAMO reads the dinamo/plugins directory and creates string representations and then instances of the objects recognized as plug-ins. In the web version, the plug-in names are passed to DINAMO via parameters in the HTML code responsible for initializing and launching the applet.

Plug-ins have two major capabilities: 1) They can perform calculations based on a sequence alignment passed to them. 2) They can create new instances of themselves that perform these calculations in their own threads of independent execution. This allows plug-ins to work in the background without interrupting foreground processes such as alignment editing.

The standard convention for calling a plug-in is straightforward. Each plug-in takes as an argument a sequence alignment object and a reference to the client

requesting the results. By evaluating the status fields in the alignment, the plug-in can determine what type of analysis is required. If the guide sequence was modified, the plug-in will need to recalculate results for the entire alignment. If any of the other sequences were modified, the plug-in will only need to calculate results for a comparison of the edited sequence versus the guide sequence.

All plug-ins perform a similar function. They analyze alignments and create categories for residues sharing similar attributes. The plug-ins all return the same information: a reference to the plug-in's legend for decoding the results, and a list of result groups containing display categories and their rank, each associated with a list of residues falling into the same category. These results are bundled so that each sequence analyzed corresponds to one result group. For example, our BLOSUM plug-in assigns matrix scores into four rankings of residue conservation. This rank may be linked to color intensity, with a more intense color indicating a conserved substitution in order to draw attention to that position. The BLOSUM plug-in therefore returns a reference to its legend object, and for each sequence analyzed: the display categories (in this case, 0, 1, 2, 3) and the residue positions in the sequence corresponding to each category.

The use of rankings allows the plug-in results to remain independent of their representation in the editor and the molecular graphics packages. Plug-ins may request that certain attributes be displayed in a particular way, but the display modules may choose to override these requests.

In order to allow components to proceed without periodically polling the plug-in for results, or worse, halting execution until the results are available, we have implemented a callback interface for the plug-ins. Any client expecting results from a plug-in first implements the plug-in's callback interface and performs the necessary update functions associated with receiving the plug-in's results in the callback routine. It is the plug-in's responsibility to call the client's version of the callback routine, passing it the results as soon as they are available. This communication link between the plug-in and the client is easily established by passing the plug-in a reference to the client expecting the results.

6. Molecular Graphics Display

The molecular graphics portion of DINAMO allows the visualization of assessment criteria in the context of the three dimensional structure of the guide protein. By mapping the results of a plug-in to a particular attribute in the molecular graphics display, such as residue color or van der Waals radius, it is possible to draw attention to regions of an alignment that may be considered unfavorable.

Communication with the graphics module takes place via a renderer-specific interpreter. The interpreter translates the display values into commands supported

by the particular molecular graphics display. Currently, interpreters exist for RasMol and Chime. Future development of renderer-specific interpreters will allow the use of other molecular graphics programs such as UCSF MidasPlus[12].

DINAMO's molecular graphics display provides a structural context for the evaluation of the alignment. For example, it allows the user to see where insertions and deletions occur in the context of the structure, and why certain alignment positions are flagged as being unfavorable. The results of several plug-ins may be viewed simultaneously by assigning a different viewing options to each one via the display mapper. For example, hydrophobicity could be mapped to color, with hydrophobic residues being colored yellow and hydrophilic residues colored blue. Residue conservation could be mapped to van der Waals sphere size, making sphere size inversely proportional to conservation, thus bringing attention to those residues that are unfavorably aligned. Each attribute can be turned on or off on a per-residue basis. Only the peptide backbone and Cβ atoms are displayed by DINAMO both in order to limit the complexity of the display and because accurate modeling of side chain conformations is difficult and computationally costly. In the case of multiple target sequences, the user may select which sequence to superimpose on the guide structure using a radio button.

In this version of DINAMO, we have chosen to use two molecular graphics engines. The web version uses the multi-platform molecular graphics program Chime[5]. Chime is a freely-available Netscape plug-in. The Netscape Navigator browser may be easily configured to include Chime, making installation simple. Currently, Chime versions exist for the Macintosh, Windows 3.1, Windows95, and SGI/Irix. The stand alone version uses a graphics interpreter for RasMol, the non-web-based predecessor of Chime which, in addition to the platforms listed above, works on virtually any Unix workstation. The display styles supported by Chime and RasMol include color, dot surfaces, labels, and space filling atoms. Also, secondary structure may be displayed as a ribbon. A RasMol/Chime command line interpreter is also included in DINAMO for experienced users who wish to make modifications to the display other than those provided by the display mapper and plug-ins. The complete list of RasMol commands is supported.

7. Summary

DINAMO is a tool for interactively building and analyzing three-dimensional models of proteins based on their sequence identity or threading onto known structures. It allows users to edit multiple sequence alignments and observe the ramifications of these adjustments instantaneously both in the sequence editor and in the molecular graphics display. DINAMO's plug-in architecture allows researchers to develop their own assessment tools and use them in DINAMO with minimal effort. The

use of display-independent plug-in results and a renderer-specific interpreter module allows DINAMO to be extended to use other molecular graphics engines by writing only the interpreter specific front ends for those packages. DINAMO is Java-based and thus highly portable and accessible to many users. It is especially well-suited to those who may need to build and analyze models only on occasion. DINAMO is available on the world wide web at http://tito.ucsc.edu/dinamo/

Acknowledgments

We thank Leslie Grate for providing the inspiration for DINAMO and for allowing us to use his in-house prototype program, SAE. We also thank Mark Diekhans for technical advice. This work was supported in part by NIH grant GM52885-R29 to LMG. Marc Hansen is supported by a GAANN fellowship.

References

1. Finkelstein, A., Protein structure: what is it possible to predict now? *Curr. Opin. Struct. Biol.*, 1997. **7**:60-71.
2. Rost, B., TOPITS: threading one-dimensional predictions into three-dimensional structures. *ISMB*, 1995. **3**:314-321.
3. Rost, B., PHD: predicting one-dimensional protein structure by profile-based neural networks. *Methods Enzymol.*, 1996. **266**:525-539.
4. Vasquez, M., Modeling side-chain conformation. *Curr. Opin. Struct. Biol.*, 1996. **6**:217-221.
5. Maffett, T., *The home page for Chime is located at: http://www.mdli.com/chemscape/chime/*, . 1997, MDL Information Systems Inc.
6. Sayle, R.A. and Milner-White, E.J., RasMol: Biomolecular graphics for all. *TIBS*, 1995. **20**:374-376.
7. Miller, R.T., Jones, D.T., and Thornton, J.M., Protein fold recognition by sequence threading: tools and assessment techniques. *Faseb J*, 1996. **10**(1):171-8.
8. Jones, D., Taylor, W., and Thornton, J., A new approach to protein fold recognition. *Nature*, 1992. **358**:86-89.
9. Abola, E.E., Bernstein, F.C., Bryant, S.H., Koetzle, T.F., and Weng, J., in *Crystallographic Databases -- Information Content, Software Systems, Scientific Applications*, F.H. Allen, G. Bergeroff, and R. Sievers, Editors. 1987, Data Commission of the International Union of Crystallography: Bonn/Cambridge/Chester. p. 107-132.
10. Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Jr., E.F.M., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T., and Tasumi, M., The

Protein Data Bank: A Computer-Based Archival File for Macromolecular Structures. *J. Mol. Biol.*, 1977. **112 (3)**:535-542.
11. Henikoff, S. and Henikoff, J.G., Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 1992:10915-10919.
12. Ferrin, T.E., Huang, C.C., Jarvis, L.E., and Langridge, R., The MIDAS Display System. *J. Mol. Graphics*, 1988. **6**:13-37.