

# Visualization of Force Fields in Protein Structure Prediction

Clark Crawford, Oliver Kreylos, Bernd Hamann, Silvia Crivelli

April 20, 2005

## Abstract

The force fields used in molecular computational biology are not mathematically defined in such a way that their mathematical representation would facilitate the straightforward application of volume visualization techniques. To visualize energy, it is necessary to define a spatial mapping for these fields. Equipped with such a mapping, we can generate volume renderings of the internal energy states in a molecule. We describe our force field, the spatial mapping that we used for energy, and the visualizations that we produced from this mapping. We provide images and animations that offer insight into the computational behavior of the energy optimization algorithms that we employ.

## 1 Introduction

A central focus in post-genomic biology is the prediction of the three-dimensional (3D) structure – the native structure – of proteins and their interactions. The 3D structures of proteins have typically been determined by means of X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy. While an increasing number of individual 3D structures are known from these experimental approaches, it is an unfortunate reality that only a small fraction of those structures have been solved due to their cost and time constraints. The need for shorter turnaround times generates great interest in more effective approaches. Among them, computational methods are a promising alternative to both complement and guide the experimental ones. Furthermore, computational methods can potentially provide insight into and understanding of the behavior of proteins on a level difficult to attain by experiments alone.

Computational methods are based on the hypothesis that the native structure of a protein corresponds to a global minimum of its free energy surface. Therefore, the protein structure prediction problem is often approached as a high-dimensional optimization problem. The objective function to be minimized can be computed by various formulae, such as CHARMM, GROMOS, ECEPP, and AMBER. Finding the global minimum of the energy surface is an extremely difficult task for several reasons:

1. the ability of energy functions to accurately model protein interactions is uncertain;
2. the number of local minima increases exponentially with the size of the protein;
3. the energy functions are ill-conditioned; and
4. the lack of effective global optimization methods that can deal with such large-scale problems.

We have implemented an energy visualization system to help researchers understand the complex biological systems they are trying to simulate. This system permits us to animate the folding process by recording the steps of an optimization procedure in terms of atom positions, energy states, and gradients. Our goal is that, through animation, we can observe changes in the force fields over time, and analyze the relationship that these fields have to a molecule's evolving structure. We can also evaluate the algorithm's behavior in comparison to expected results, and monitor its progress.

Because the objective function assigns a single scalar value to an entire protein, it is difficult to visualize the relationship between the energy function and protein structure in an effective way. We use a straightforward calculation to map selected components of the objective function back to the positions of the atoms comprising the protein, enabling volume visualization techniques to display the two in superposition. These combined visualizations lead to a better understanding of both the objective function and the ongoing optimization process.

The energy visualization system is implemented in conjunction with the energy computation plug-in architecture of the **ProteinShop** application software [3, 5, 6, 7]. **ProteinShop** is a graphical environment developed to create low-energy structures for use as initial configurations in a global optimization process. Therefore, it supports on-the-fly calculation of a protein structure's internal energy using the same function used by the global optimization algorithm. This feature allows users to judge the overall quality of the structures generated. To be useful in a more general context, **ProteinShop** provides a plug-in system that allows users to specify their own energy formulations.

Integration with **ProteinShop** allows the energy visualization to be utilized in conjunction with the expanding set of steering and analysis features in that application. Use of the plug-in architecture will make possible the comparative analysis of different energy computation formulae and optimization algorithms on specific inputs. We expect that the pending release of **ProteinShop** under an open-source license will facilitate more rapid expansion of the repertoire of algorithms that are available in its plug-ins.

In the future, **ProteinShop**'s visualization of molecular force fields will be applicable to more than protein folding applications. It will also assist in analysis of molecular docking and the stability of multiple-protein structures. The visualization system only requires the ability to measure force fields in relation

to the positions of atoms, residues, and secondary structures. As capabilities are added to the calculator and optimization systems, this visualization system will support them. Moreover, this visualization approach can find application in the analysis of high-dimensional optimization problems in general.

## 2 Related work

**ProteinShop** (Figure 1) was originally designed to support a protein structure prediction method involving several members of our group [3]. This method is based on two phases. The first phase generates initial structures, which are local minima. The second phase improves upon the initial structures using both global and local minimizations. Because there is no global optimization algorithm that can deal with the large number of variables involved in this type of problem, the global optimization phase improves the initial configurations through global optimizations in subspaces of the full-dimensional space. One advantage of this approach is that it can be parallelized by selecting different subsets of dihedral angles and performing small-scale global optimizations on those subsets. Those small-scale global optimizations produce a number of minima in the chosen subspaces. A number of those conformations are selected for local minimizations in the full variable space. The new local minima are merged into a list of possible solutions ordered by energy value. The process repeats iteratively until no further lowering of energy is observed between consecutive iterations. The global optimization process can be viewed as a search through a large tree of possible solutions. Each node of this tree corresponds to a local minimum and its children nodes to the local minima generated from it by performing global optimizations on a subset followed by local minimizations of the full dimensional space.

**ProteinShop** provides support for the first phase of the protein structure prediction method. Guided by the energy function, it quickly creates a variety of protein configurations and locally minimizes them to find low-energy candidates for the global optimization phase. To that end, it includes a plug-in to compute the AMBER energy of a protein (see Section 3.1) and to perform local minimization of this energy. The local minimizations are performed using the Limited Memory BFGS algorithm (LBFGS), as implemented in the OPT++ toolkit [9], running interactively inside the **ProteinShop** window. In this context, our energy visualization system allows real-time visualization of the protein minimization process that drives the protein to its local minimum with the goal of studying, analyzing, and comparing energy functions as well as local minimization algorithms.

**ProteinShop** also supports the second phase of the structure prediction method by providing a graphical environment to monitor and steer the global optimization process. **ProteinShop** supports interaction with the configuration and subspace selection module of the global optimization process while it is running and provides access to its internal data structures. By using this data, **ProteinShop** can create a graph of the entire tree of possible configurations

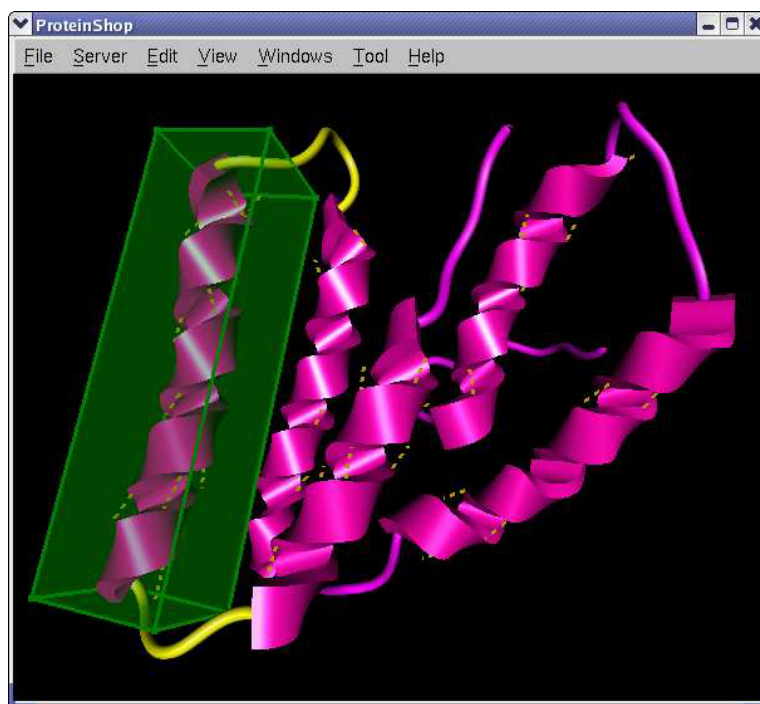


Figure 1: ProteinShop.

generated by the global optimization process thus far and make them accessible for viewing and manipulation by the user. The user can locally optimize the manipulated structure and insert it back into the global optimization process. The idea is that a knowledgeable researcher who is following the global optimization process can make changes to certain structures, returning them to an energy-decreasing path. In this context, the energy visualization system allows users to analyze important information such as which configurations are forming hydrophobic cores and which areas of a configuration are more likely to produce a larger drop in energy, making them good candidates for further minimization. The energy visualization system helps users focus the search on the most promising areas of the tree, thus reducing the time needed to find a solution.

### 3 Force field visualization

The energy visualization system renders the force fields as a semitransparent cloud around the various geometric “tinkertoys” that can be used to display the molecule’s structure. Where the cloud is thickest, the forces are strongest.

Where the cloud is wispy or nonexistent, the forces are reaching equilibrium. Rendering is straightforward, done by hardware with volume textures. The user controls the resolution detail of the texture and all important aspects of the transfer function, which is tailored to `ProteinShop`'s functionality.

Section 3.1 describes the force field calculator implemented in `ProteinShop`'s AMBER plug-in. Section 3.2 describes the pipeline for the energy visualization. Although we only look at AMBER in this context, other force fields can be visualized for comparative or analytical purposes by changing the plug-in.

### 3.1 AMBER

The AMBER force field (Assisted Model Building with Energy Refinement) is used to evaluate the stability of the molecule in response to local changes in its configuration produced by the modeling tools in `ProteinShop`. The configuration of the molecule is defined by the positions of its atoms. The terms of the force field are defined by the differences between the states of local elements in the configuration (bond angles, distances, etc.) from locally defined equilibrium values. The greater the difference, the higher the energy. When the energy is minimized, the molecule is assumed to be in a stable state.

The force field definition consists of five terms, which can be visualized individually. The force field definition is based on [10]:

$$\begin{aligned}
 E_{total} = & \sum_{bonds} K_R(R - R_0)^2 + \\
 & \sum_{angles} K_\theta(\theta - \theta_0)^2 + \\
 & \sum_{dihedrals} \frac{K_\phi}{2} [1 + \cos(n\phi - \gamma)] + \\
 & \sum_{nonbonded\ pairs\ i,j} \left[ \frac{A_{i,j}}{R_{i,j}^{12}} - \frac{B_{i,j}}{R_{i,j}^6} + \frac{q_i q_j}{\epsilon_r \epsilon_0 R_{i,j}} \right].
 \end{aligned}$$

In the following, we discuss the meaning of the various variables appearing in this formula. The formulation shows only four terms; we produce an additional nonbonded term for certain pairs of atoms that are separated by exactly three bonds, called "1-4 nonbonded energy". To visualize these energies we map them back to locations in space, averaging them in a limited volume that is concentrated around the positions of the contributing atoms (two atoms for bonded and nonbonded pairs, three atoms for angles, and four atoms for dihedrals). These terms are illustrated in Figure 2.

We visualize the force field terms individually to attain a better understanding of the relative influence exerted by different terms. For this reason, we refer to the energy terms associated with the  $i^{th}$  atom and their gradient vectors on an individual basis as:

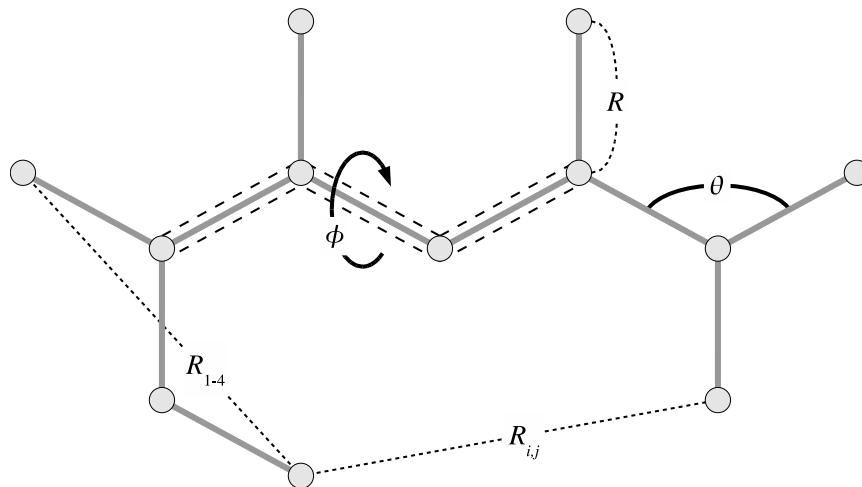


Figure 2: Visualized optimization targets in the AMBER formulation: bond radius  $R$ , bond angle  $\theta$ , dihedral angle  $\phi$ , nonbonded radius  $R_{i,j}$ , and 1-4 nonbonded radius  $R_{1-4}$ . There are actually two dihedral angles along the backbone of each residue, called  $\phi$  and  $\psi$  (not shown).

- *Bond*:  $A_1^i$  and  $\nabla \bar{A}_1^i$ .
- *Angle*:  $A_2^i$  and  $\nabla \bar{A}_2^i$ .
- *Dihedral*:  $A_3^i$  and  $\nabla \bar{A}_3^i$ .
- *1-4 Nonbonded*:  $A_4^i$  and  $\nabla \bar{A}_4^i$ .
- *Full nonbonded*:  $A_5^i$  and  $\nabla \bar{A}_5^i$ .

The gradients are based on the first derivative of the AMBER formulation.

### 3.2 Energy rendering

The energy rendering system is built on top of ProteinShop's older energy visualization feature [3], which remains available to users. In particular, the controls for that system are also used by the new system. Including both the original settings and the new ones added for this system, the user has a total of 8 settings to control the transfer function and determine the general appearance and information conveyed by the energy cloud. The assemblage of these settings is in Figure 3.

1. **Channel**: The user can show either the subset sum of the energy terms selected in the discriminator, or the subset sum of their gradient magnitudes.

2. **Discriminator:** This is a block of toggles in the user interface through which the user can select an arbitrary subset of the energy component terms to be visualized. Those not selected will be ignored. This setting and setting 3 (clamp) were part of `ProteinShop`'s original energy visualization functionality.
3. **Clamp:** This interval helps the user eliminate outliers from the data, which might otherwise hide detailed information elsewhere.
4. **Resolution:** The user can set the resolution in texels per angstrom ( $\text{\AA}$ ). The selected resolution may be automatically lowered to observe constraints imposed by the platform's physical memory and OpenGL rendering capabilities.
5. **Radial specifier:** The user specifies a multiplier and coefficient type for the radial basis function. The coefficient type can be either uniform ( $\times 1\text{\AA}$ ) or relative; in the latter case equal to either of each atom's physical or Van der Waals radius. The final radius is defined in  $\text{\AA}$ .
6. **Classifier:** The user can specify one classification function, which maps atoms to a limited range of integers  $[0, m)$ , where  $m$  is the number of classifications in the function's range. The classifier's domain consists of everything `ProteinShop` knows about the atoms, including their element types, positions, topological relationships, current force field states, and the secondary structures and residues that they belong to.
7. **Normalizing interval:** This interval determines how the cumulative atom energy values from the input channel are normalized into the domain of the color function (below). It can be computed automatically based on the current energy levels or set to an arbitrary value.
8. **Color function:** Each integer in the classifier's range is associated with a color function. The color function maps the atom's energy to a color. The colors from all classifications are combined in a weighted average to produce the final color and transparency of the texture.

The atom energy store in Figure 3 is the AMBER plug-in, which provides real-valued energy component terms and gradient vectors for each atom in the molecule. These numbers are processed according to the channel selected to produce a single floating point value for each atom. Only the component terms selected in the discriminator are included. If no terms are selected in the discriminator, every atom's value will be zero. The number of toggles in the discriminator,  $c$ , is determined by the plug-in. For our AMBER plug-in,  $c = 5$  for the terms illustrated in Figure 2. If, for example, a solvation term is added to the force field, it will appear in the user interface as a sixth toggle in the discriminator.

Let the discriminator function  $D(j) = 1$  if the  $j^{\text{th}}$  energy component is selected and 0 if not,  $0 \leq j < c$ . We compute the value  $e_i$  of the  $i^{\text{th}}$  atom as

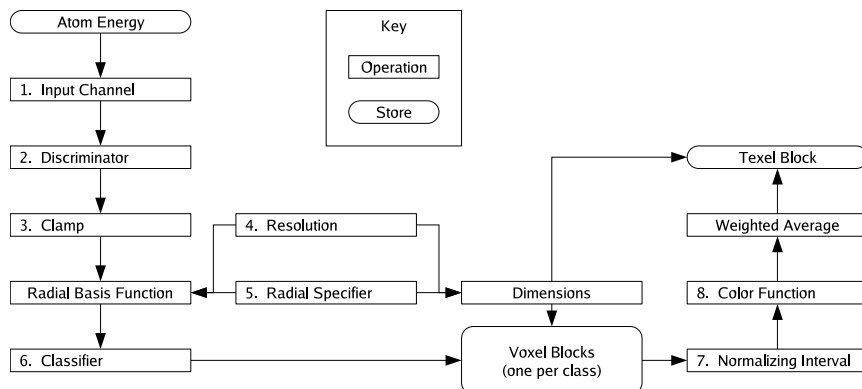


Figure 3: Energy visualization pipeline used in ProteinShop.

$$e_i = \sum_{j=0}^{c-1} D(j) \cdot \left\{ \begin{array}{ll} A_j^i & \text{for subset sum} \\ \|\nabla A_1^i\| & \text{for gradients} \end{array} \right\}. \quad (1)$$

The value of  $e_i$  is then clamped, and spread through the texel block by means of the radial basis and classification functions. The radius of the basis function  $s$  is determined by the radial specifier, equal to the product of a multiplier chosen by the user with a slider and one of three coefficients: a constant (chosen with another slider), the atom’s radius, or the atom’s Van der Waals radius. The basis function  $f(r_i)$  is a smooth curve similar to those used for the implicit modeling of molecular surfaces [1]. It depends on the texel’s distance  $r_i$  from the center of each atom:

$$R(r_i) = \left\{ \begin{array}{ll} 1 - \frac{3r_i^2}{s^2} + \frac{2r_i^3}{s^3} & \text{if } r_i < s \\ 0 & \text{otherwise} \end{array} \right\}. \quad (2)$$

The voxel block store holds texel magnitudes for each classification. Let the classification function  $L(i, k) = 1$  if the  $i^{\text{th}}$  atom belongs to the  $k^{\text{th}}$  classification and 0 if not;  $0 \leq k < m$  and  $0 \leq i < n$ , where  $n$  is the number of atoms in the molecule. Given the atom energy value  $e_i$  (1), the radial basis  $R(r_i)$  (2), and the classifier  $L(i, k)$ , the texel magnitude  $t_k$  becomes

$$t_k = \sum_{i=0}^{n-1} e_i \cdot R(r_i) \cdot L(i, k). \quad (3)$$

The normalizing interval  $N(t_k)$  maps texel magnitudes to the unit interval (clamp and scale) for use with color functions. The color function  $C(N(t_k))$  implements an arbitrary continuous color map. ProteinShop provides a dozen of these, including intensity functions (ranging from a component color at 0 to white at 1 through different paths), constant functions, and invisibility to hide



selected parts of the molecule. The final texel color  $t$  is computed from the classified texel magnitudes  $t_k$  (3) as a weighted average, defined as

$$t = \frac{\sum_{k=0}^{m-1} N(t_k) \cdot C(N(t_k))}{\sum_{k=0}^{m-1} N(t_k)}. \quad (4)$$

## 4 Results

It is possible to implement this pipeline in  $O(n \cdot (s \cdot q)^3)$  time, where  $q$  is the resolution of the texture grid, by classifying each atom and determining which portion of the texture grid it will affect prior to iterative computation of (3). The pixel transfer operations will require  $O(N^3)$  time in the size of the texel block regardless, but hardware makes this part of the computation relatively fast. In practice, depending on the size of the molecule and the resolution chosen, the pipeline takes anywhere from a fraction of a second to half a minute or more, but all of the textures seen here were produced in less than ten seconds on an obsolete machine (Pentium III, 733 MHz) with no 3D texture capability at all. Once generated, the textures can be viewed at interactive refresh rates, given suitable graphics hardware.

Three classifiers were implemented to demonstrate the system. The default classifier is called the unity function, defined as  $L(i, 1) = 1, \forall i \in [0, n)$ . The configuration shown in Figure 4 was locally optimized inside `ProteinShop` by our energy plug-in. A playback feature is available that records the state of each iteration in the minimization to a binary file, allowing later analysis and review. This feature can be used to produce animation frames, or simply to flip back and forth between selected states in order to produce images like these, which show the sum of the AMBER energy terms for each atom before and after minimization.

The second classifier distinguishes atoms belonging to dipoles forming hydrogen bonds from the others. Figure 5 shows two views of 1p9x made with this classifier that are identical except in their energy rendering. The utility of the invisible color function is demonstrated by its use in this case, because the dipole atoms are small in number. The force fields of atoms from small classes can be overwhelmed or obscured by large numbers of atoms in other classes.

The third classifier distinguishes atoms belonging to hydrophobic residues from those belonging to hydrophilic residues, and both of these from atoms whose residues are neither hydrophobic nor hydrophilic. A larger radial specifier was used in Figure 6 to obtain a better understanding of the overall shape of the molecule. This classifier can be used to evaluate the effects of solvation terms in the force field.

## 5 Conclusions and future work

The classifiers and color functions are implemented in a highly modular way that makes the process of adding new functions to the source code and user interface

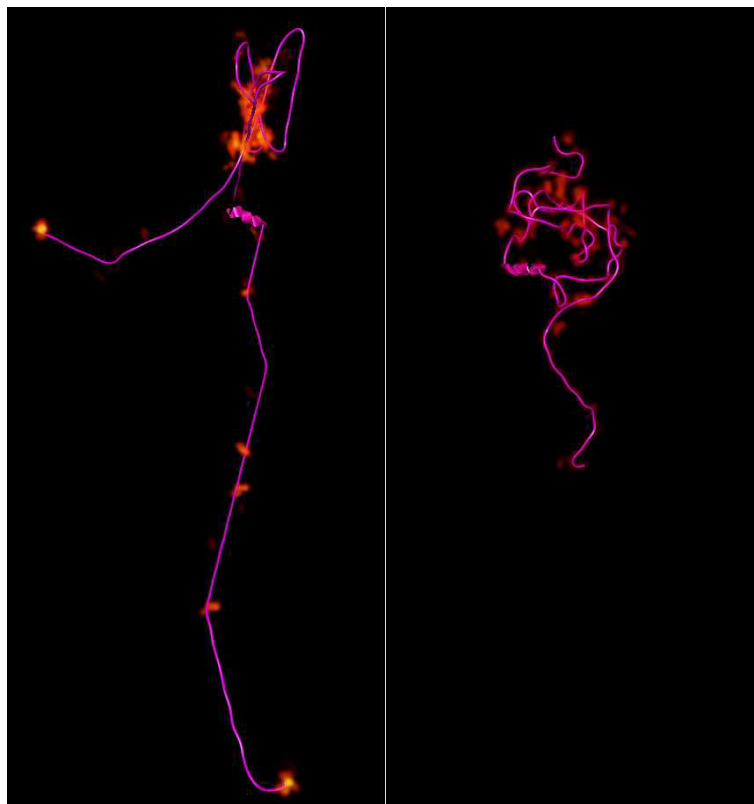


Figure 4: Configurations of CASP6 target 209 before and after local minimization inside `ProteinShop`. The intensity of color shows the relative magnitude of the AMBER energy terms for each atom.

very simple. The actual time required depends on the complexity of the function, but a rich set of classifiers can easily be created based on `ProteinShop`'s existing functionality. Scientists may also find it useful to build data mining tools on this framework. Such a system would exploit existing hooks in the framework to create instantiable functions that can be edited by the user through a customized user interface. As a simple example, a classifier might be edited through a checkbox list of element names, and in this manner form a two-set partition of the elements. As a more complex example, the editor of a compound classifier might allow the user to specify one input classifier, and then associate each element of that input's range with another classifier.

To support future analyses of protein docking and interaction, the rendering system must be expanded to support the force fields of multiple molecules. This will also require work elsewhere in `ProteinShop`, but new classifiers to support

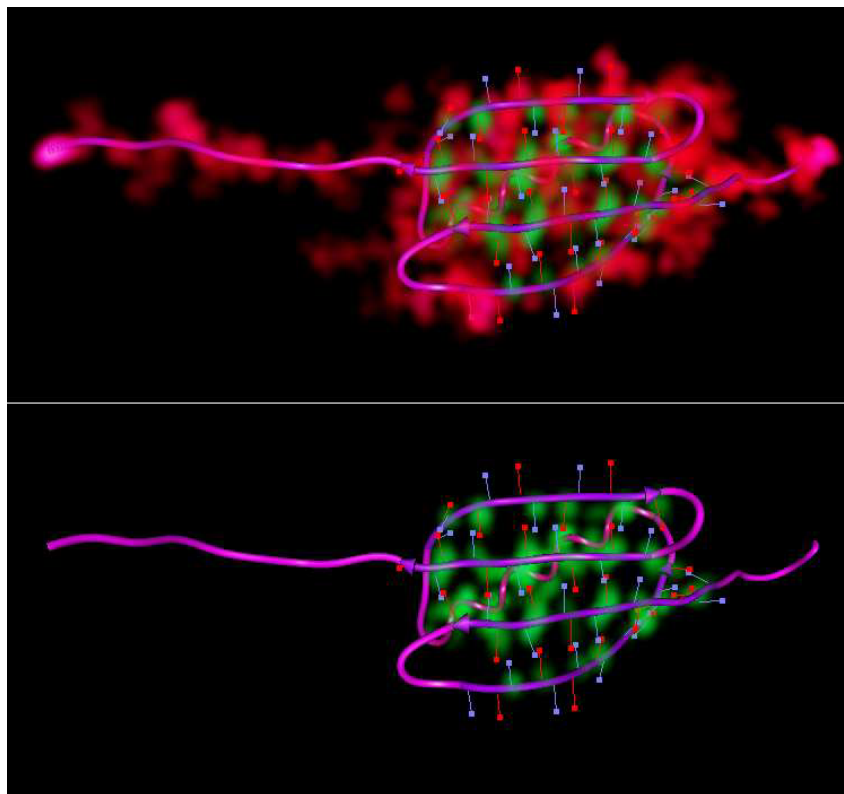


Figure 5: Two views of 1pgx showing gradients over hydrogen bond sites. Above, atoms that belong to bonded dipoles are green; all other atoms are red. Below, atoms not belonging to bonded dipoles are hidden.

docking analysis will be needed. For example, a docking classifier might distinguish atoms dominated to varying degrees by inter-molecular forces from those that are not. This functionality would be highly dependent on the calculator plug-in, which is another general area that calls for additional work. Additional plug-ins will allow different force field formulations to be comparatively analyzed on a visual basis.

## 6 Acknowledgments

Ricardo Oliva built the local optimization solver based on Newton's method (LBFGS) and the AMBER evaluator described in section 3.1.

This work was also supported by the National Science Foundation under contract ACI 9624034 (CAREER Award), through the Large Scientific and Soft-

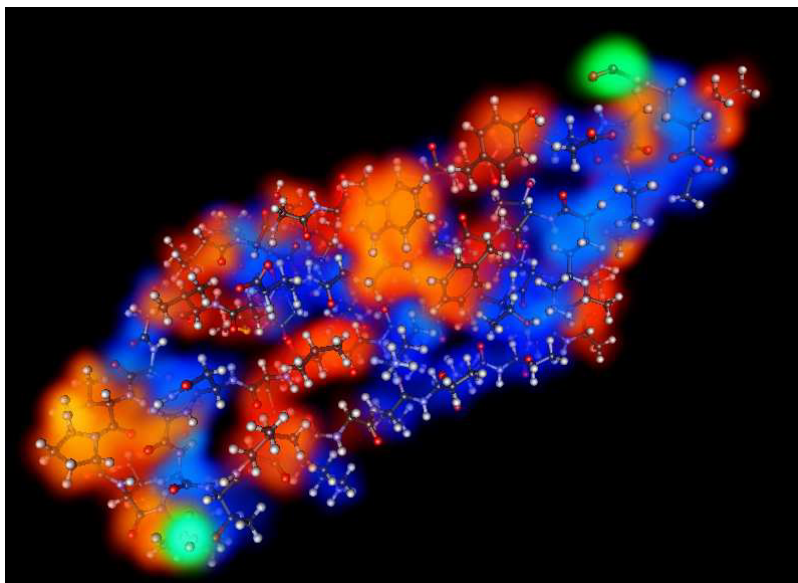


Figure 6: A different configuration of 1pgx showing gradients over CPK geometry. The radial specifier is 1.5 times the size of the Van der Waals radius. Atoms belonging to hydrophilic residues are blue, hydrophobic orange, and unclassified residues at the ends of the chain are green.

ware Data Set Visualization (LSSDSV) program under contract ACI 9982251, through the National Partnership for Advanced Computational Infrastructure (NPACI) and a large Information Technology Research (ITR) grant; and the National Institutes of Health under contract P20 MH60975-06A2, funded by the National Institute of Mental Health and the National Science Foundation. We gratefully acknowledge the support of the W.M. Keck Foundation provided to the UC Davis Center for Active Visualization in the Earth Sciences (CAVES), and thank the members of the Visualization and Computer Graphics Research Group at the Institute for Data Analysis and Visualization (IDAV) at the University of California, Davis.

## References

- [1] James F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, 1982.
- [2] S. Crivelli, E. Eskow, B. Bader, V. Lamberti, R. Byrd, R. Schnabel, and T. Head-Gordon. A physical approach to protein structure prediction. *Biophysics Journal*, 82:36–49, 2000.

- [3] Silvia N. Crivelli, Oliver Kreylos, Bernd Hamann, Nelson Max, and E. Wes Bethel. ProteinShop: A tool for interactive protein manipulation. *Journal of Computer-Aided Molecular Design*, 18:271–285, 2004.
- [4] Oliver Kreylos. *Interactive Visualization and Computational Steering*. PhD thesis, University of California, Davis, 2002.
- [5] Oliver Kreylos, Nelson Max, and Silvia Crivelli. ProtoShop: Interactive design of protein structures. In J. Moult, K. Fidelis, A. Zemla, and T. Hubbard, editors, *Proceedings of CASP5 - Fifth Meeting on the Critical Assessment of Techniques for Protein Structure Prediction*, pages A213–A214, Pacific Grove, California, December 1-5 2002.
- [6] Oliver Kreylos, Nelson Max, Bernd Hamann, Silvia N. Crivelli, and E. Wes Bethel. Interactive protein manipulation. In *Proceedings of IEEE Visualization*, pages 581–588, 2003.
- [7] Lawrence Berkeley National Laboratory. ProteinShop home page. <http://proteinshop.lbl.gov>.
- [8] D. C. Liu and J. Nodcal. On the limited memory BFGS method for large scale optimization methods. *Mathematical Programming*, 45, 1989.
- [9] Juan C. Meza. OPT++: An object-oriented class library for nonlinear optimization. Sandia Technical Report SAND94-8225, Sandia National Laboratories, Livermore, CA, March 1994.
- [10] S. J. Weiner, P. Kollman, D. Case, U. Singh, C. Ghio, G. Alagona, S. Profeta, and P. Weiner. A new force field for molecular mechanical simulation of nucleic acids and proteins. *Journal of the American Chemical Society*, 106(765), 1984.