Brian Burns
BCHS 4312 Modeling Biological Macromolecules
Dr. Briggs
May 9, 2005

Protein-DNA Docking by Iterative Refinement

Abstract

DNA-binding proteins are important in regulating the transcription of genes, and understanding the interactions of protein and DNA will be important in building towards parameterized models of these molecules. Rigid-body docking methods can be used to find the best configurations of a given protein and DNA molecule, based on shape and electrostatic complementarity. The program FTDock [2] performs a global search over all possible configurations, but this requires a large amount of computer time, or access to parallel computers. To make this approach more accessible, the FTDock program was adapted for iterative refinement, by starting at low resolutions, finding the best configurations, then exploring around these at a higher resolution. The results show that this approach cuts down on the computer time considerably, though at the grid sizes used the program was unable to locate the correct binding configuration.

Introduction

In modeling physical systems, it is usually a good approach to start with the simplest possible model, then slowly add complexity as needed. To that end, modeling protein-DNA interactions by low-resolution models would seem to be a good approach.

The antennapedia-DNA complex (PDB **9ANT** [1], resolution 2.4A) provides a simple model for protein-DNA interactions, since it is a small complex and does not involve too much bending of the DNA molecule. It will be used for this test of iterative search for rigid-body docking.

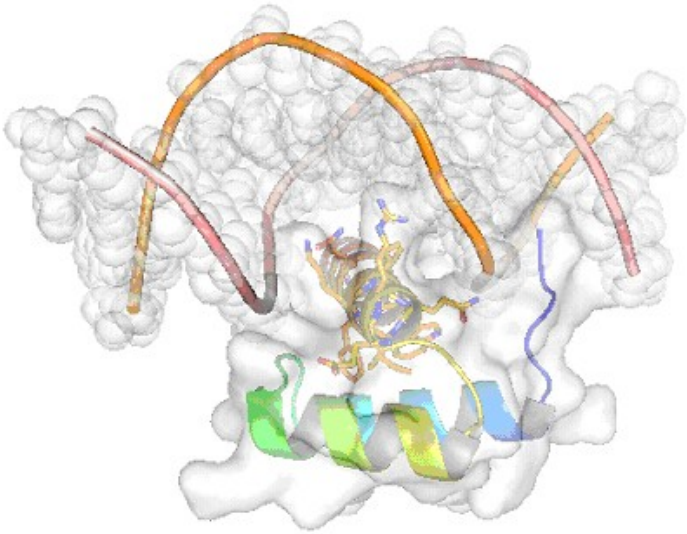

Fig. 1. Antennapedia-DNA complex (9ANT) The third helix fits right in the major groove of the DNA, while the N-terminus sits in the minor groove.

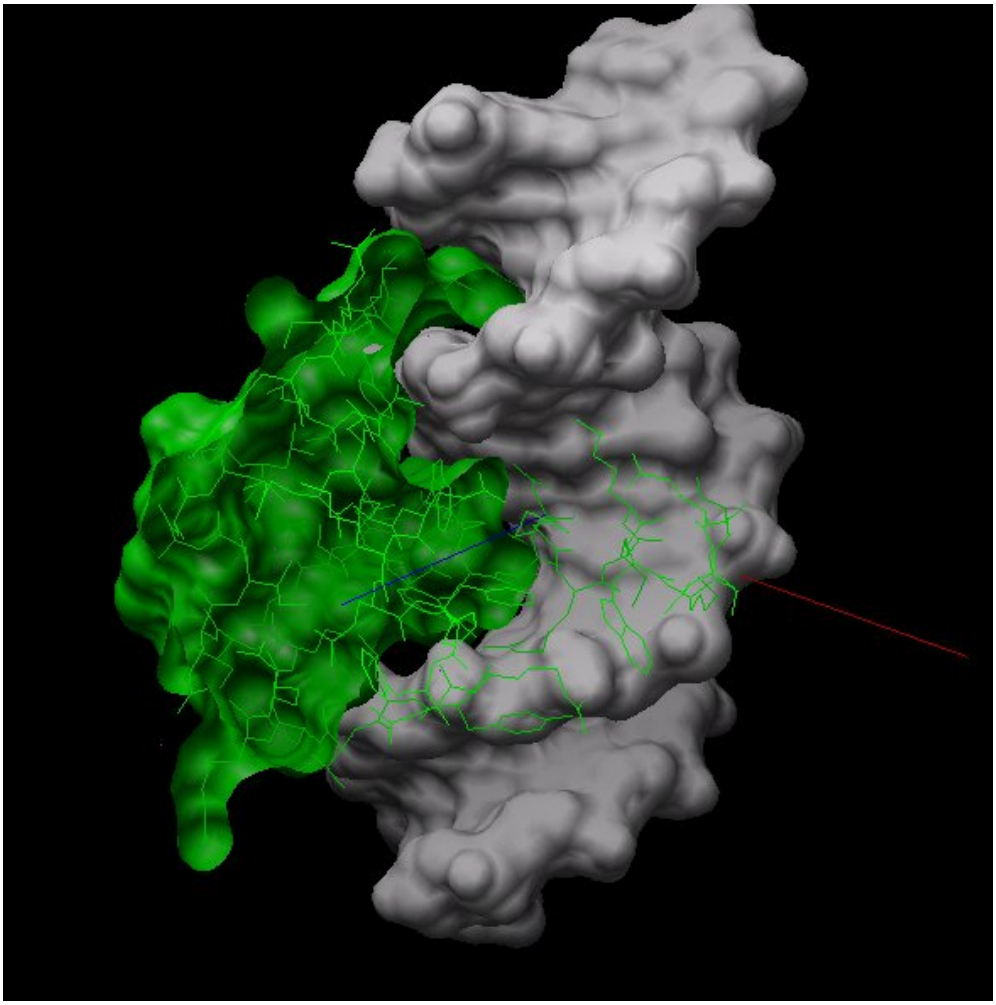

Fig. 2. Surfaces with cutaway image (made using **Chimera/Per Model Clipping** module)

The program **FTDock** [2] handles rigid-body docking using shape and (optionally) electrostatic complementarity, using FFT and convolution to speed the searching (going from N^2 to N logN). It is open source and written in C, and proved to be fairly easy to adapt. It was modified so that it could be called by a wrapper program, written in Python, which calls it at progressively higher levels of resolution, exploring around the best configurations between iterations.

Note that a full literature search was not performed, due to time limitations and the large number of references to evaluate, so there may be other, better approaches out there, or I may be duplicating someone else's efforts.

Methods

The following procedures were performed, using various programs:

Method	Program	Version
Energy Minimization	Insight II / Discover	2000.1
Electrostatic Potential Surfaces	Grasp	
Rigid-Body Docking	FTDock	2.0
Iterative Rigid-Body Docking	Customized FTDock	

Most images included in this report were generated with **Chimera**, and include a set of xyz axes for reference (colored rgb, respectively) which are each 25 angstroms long.

PDB Files Used

A standard set of pdb files was constructed from the original crystal structure, for use with the various experiments. They include the following:

PDB file	Description
9ant_orig	original pdb (9ANT.pdb) with two versions of the protein-dna complex. not used.
9ant	removed 2nd structure (chains B,E,F), Ni atom, rotated complex so that DNA aligns with y-axis, centered on DNA center of mass. includes waters, some of which are in interface between DNA and protein. split to antp.pdb and dna.pdb.
9ant_dry	same but no water
antp	protein alone with water
antp_dry	protein alone with no water (a little harder to dock)
antp_em	energy minimized version, no water (hardest to dock)
dna	the dna from the crystal (slightly bent)
dna_em	straight dna version (harder to dock)

Energy Minimization

Energy Minimization was performed on the antennapedia protein, producing the file antp_em.pdb. Several runs were done, starting with Steepest Descent for 1000 iterations, then proceeding to Conjugate Gradient with a layer of water 5 angstroms thick surrounding the protein, for a total of 20,000 iterations.

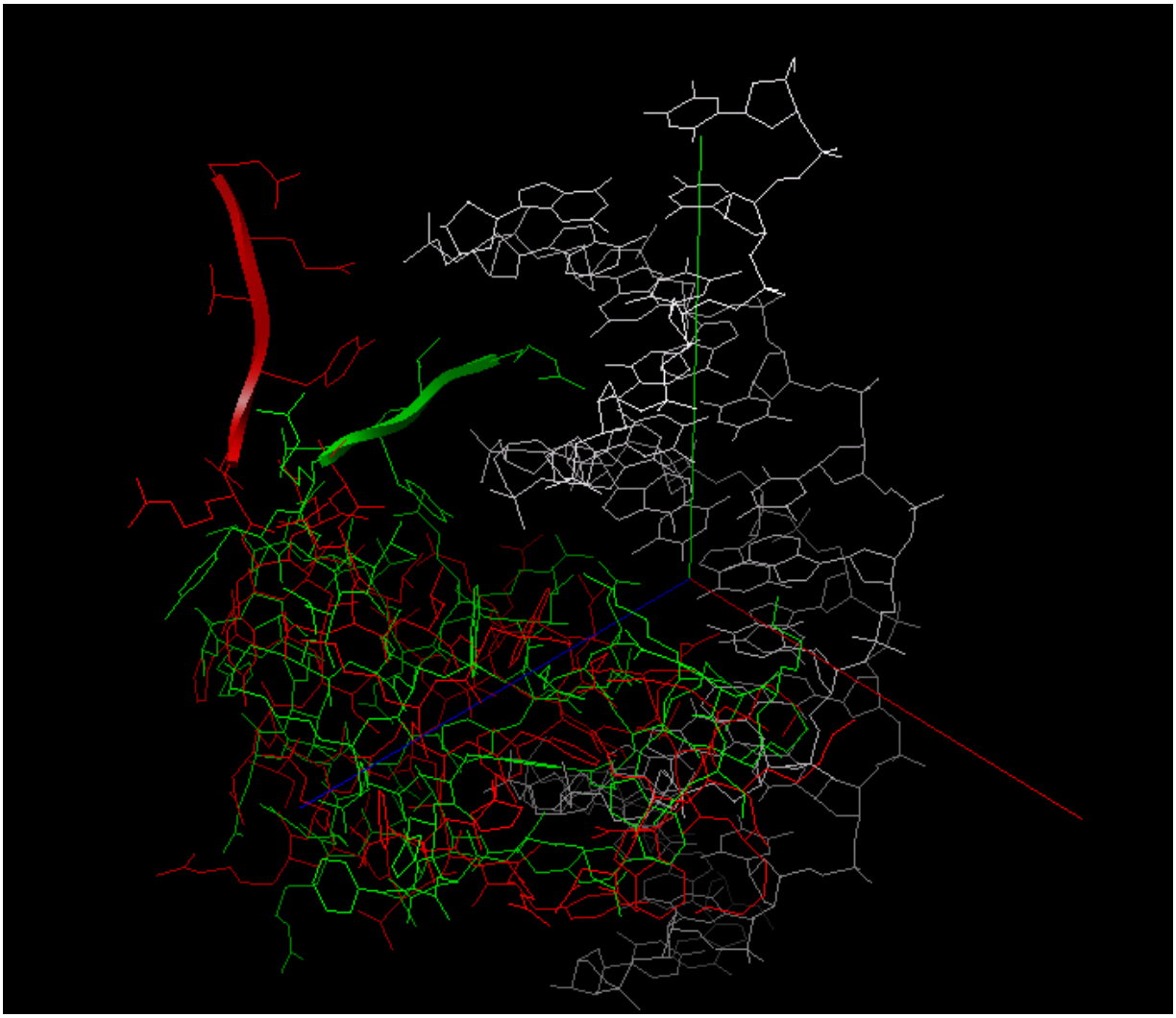

Fig. 3. Original protein (green) and energy minimized version (red), showing large relaxation of N-terminus. RMSD between 56 atom pairs of 4.306 angstroms.

The energy minimized DNA is actually just a straight molecule built using **Biopolymer** in the **InsightII/Builder** module, with the same base-pair sequence as the crystal (AGAAAGCCATTAGAG). Hydrogens were then deleted and chain ID's were added to the pdb file (ie chain C and D, to match 9ant.pdb). The molecule was then aligned to match the structure in dna.pdb by using the **Chimera/MatchMaker** module. Then half a base pair (15:C) and extra atoms were trimmed from the new molecule to make it match the dna.pdb structure, and a single base (T) was added.

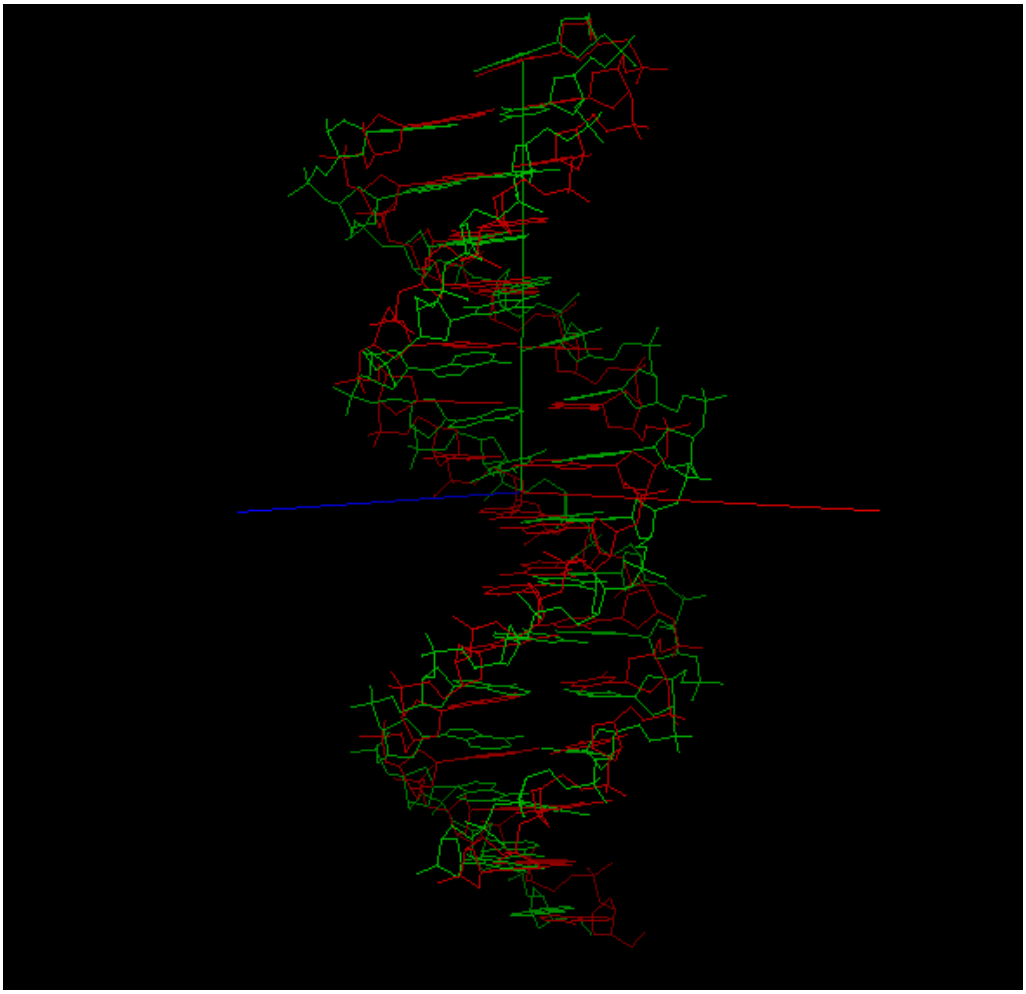

Fig. 4. Original DNA (green) and "energy-minimized" version (red). RMSD between 30 atom pairs is 2.615 angstroms.

Electrostatic Potential Surfaces

The DelPhi module of InsightII was used to create potential grids for each molecule, then their surfaces were colored according to the potential. The results are shown in Figure 5, with the molecules slightly removed from one another. I'm not sure if the DNA coloration is correct - it seems odd that one side of it would have such a different potential than the other, though if this is not a mistake it might possibly be due to the bending of the DNA.

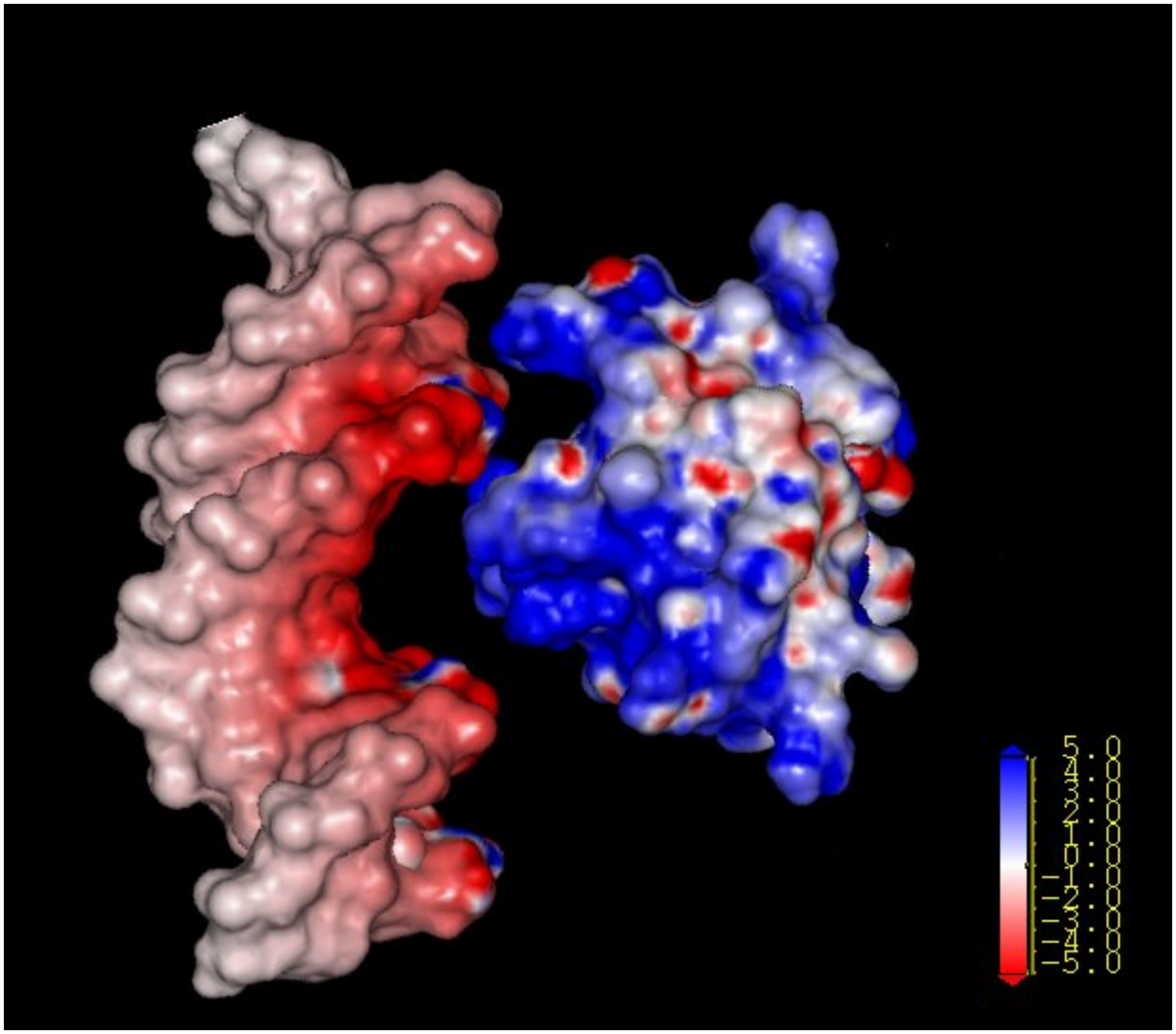

Fig. 5. Molecular surfaces colored by electrostatic potential (built using InsightII).

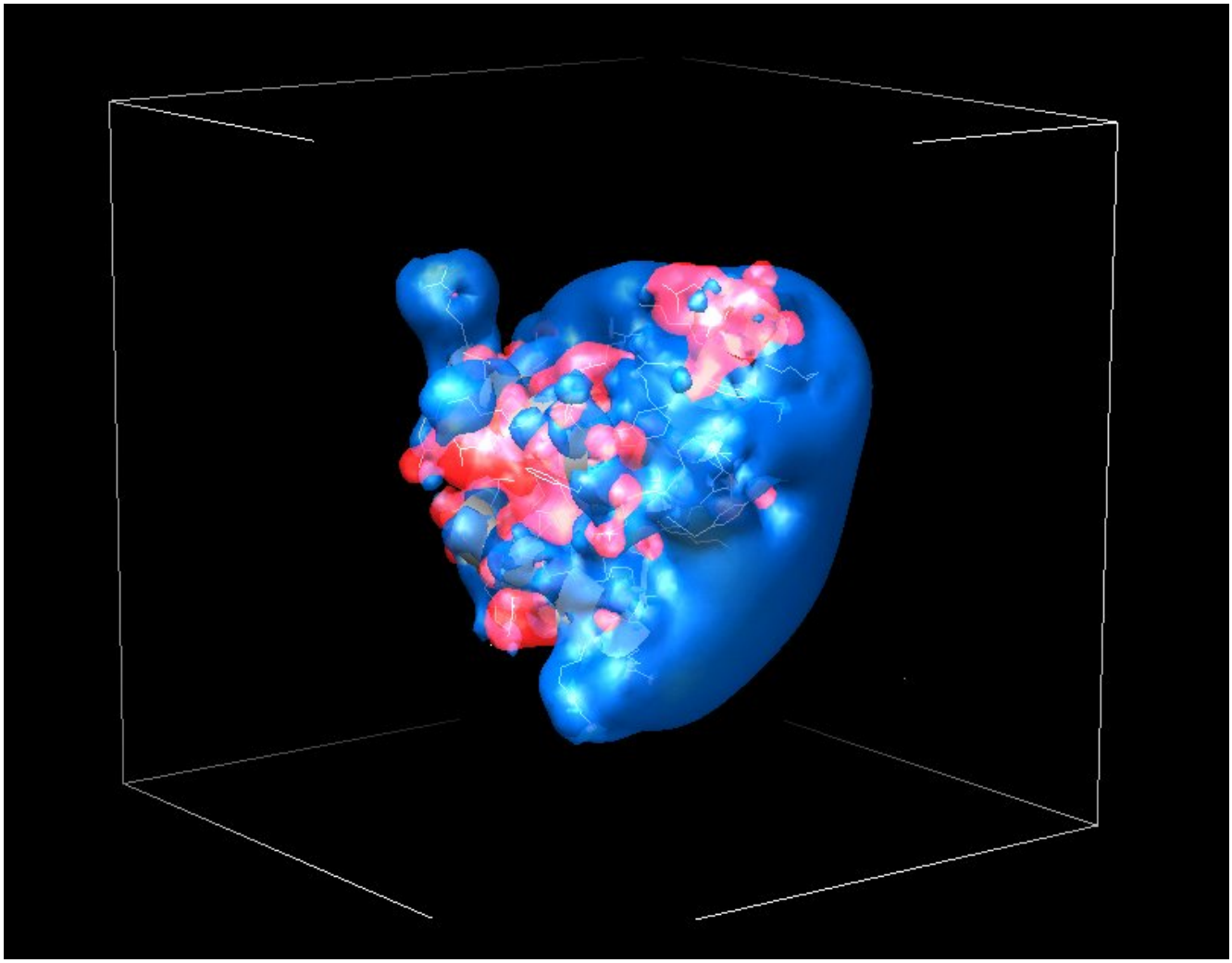

Fig. 6. Isosurfaces of electrostatic potential around antennapedia protein. Built using Chimera/Volume Viewer.

FTDock

Gridsize must be integer and even (for FFT), lowest gridsize should give a resolution of about 1 angstrom (due to FTDock using integer FFT routines?). Default anglestep is 12 degrees. Trying to go below 5 degrees will give the error "I refuse to do that many angles."

Ran FTDock with default parameters (gridsize 130, anglestep 12 degrees, noelec, total of 9240 angles to evaluate) - took **58 hours**! Correct orientation of (0,0,0) was not even in the top 10,000 scores (cursory search, not accounting for possible nearby hits).

Customized FTDock

Modified FTDock to allow it to be called iteratively from another program, in this case a Python program that specifies the grid sizes and angle steps to iterate over. Also needed to limit the best configurations used to keep the protein from sticking onto the ends of the DNA. To do that had to rotate the original PDB file so the axis of the DNA was aligned with the y-axis, then allow only configurations where abs(y) was within some limit (chose 20 angstroms).

In general, wrap.py uses the current directory to store files for each iteration, then copies them to a subfolder named 'run'. After done with a run this folder can be renamed, eg to 'run0507', or 'run_em_elec'. The program will also copy itself into the run folder, since it's a good idea to keep the code with the data generated (in lieu of having a well defined input parameter file). It also outputs run information to a logfile so you can keep track of when you did what. The contents of a typical run folder are shown below. For each iteration performed it includes the best angles, the best configuration of the protein in a pdb file, the best configuration of the protein and dna in a pdb file, the logfile, a VRML file with colored cones to indicate the location and orientation of the different fits, the score files produced by FTDock, wrap.py, and view.py.

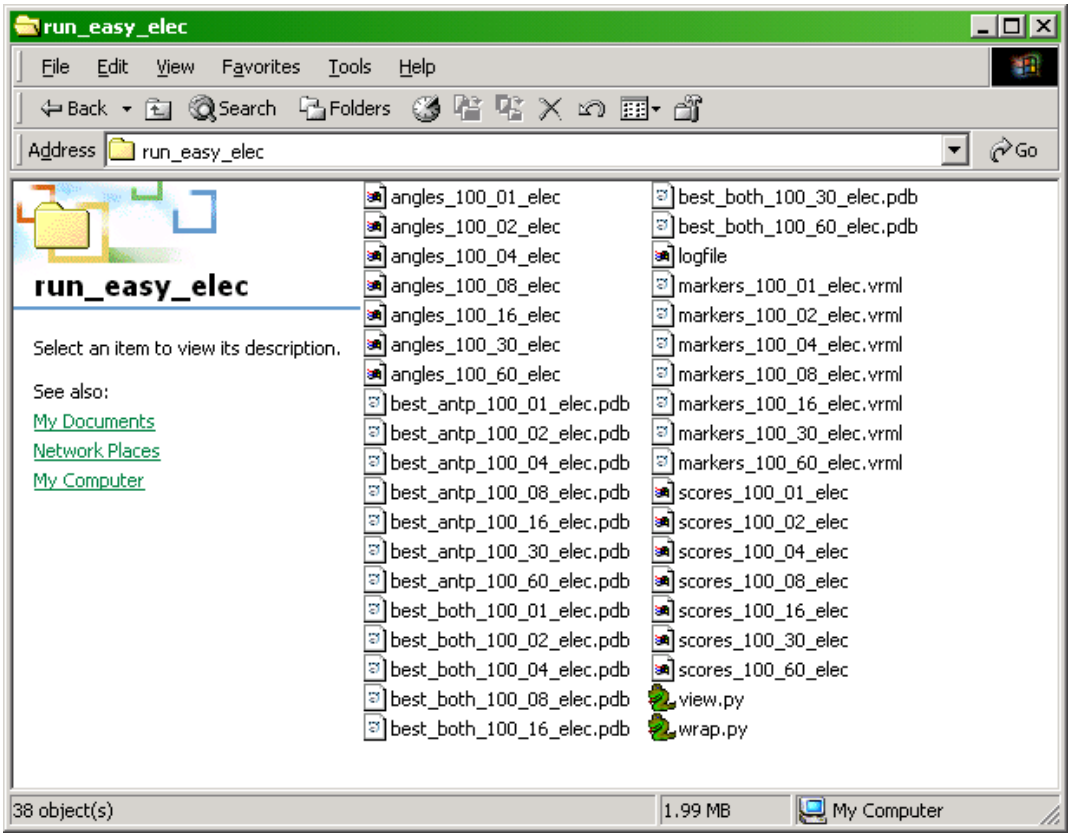

Fig. 7. Contents of a typical run folder.

View.py is a Chimera code file that loads the original crystal structure, the best protein configuration for some iteration, xyz axes, the VRML cone markers, calculates an RMSD, and orients the complex to two good views and takes pictures (as png files).

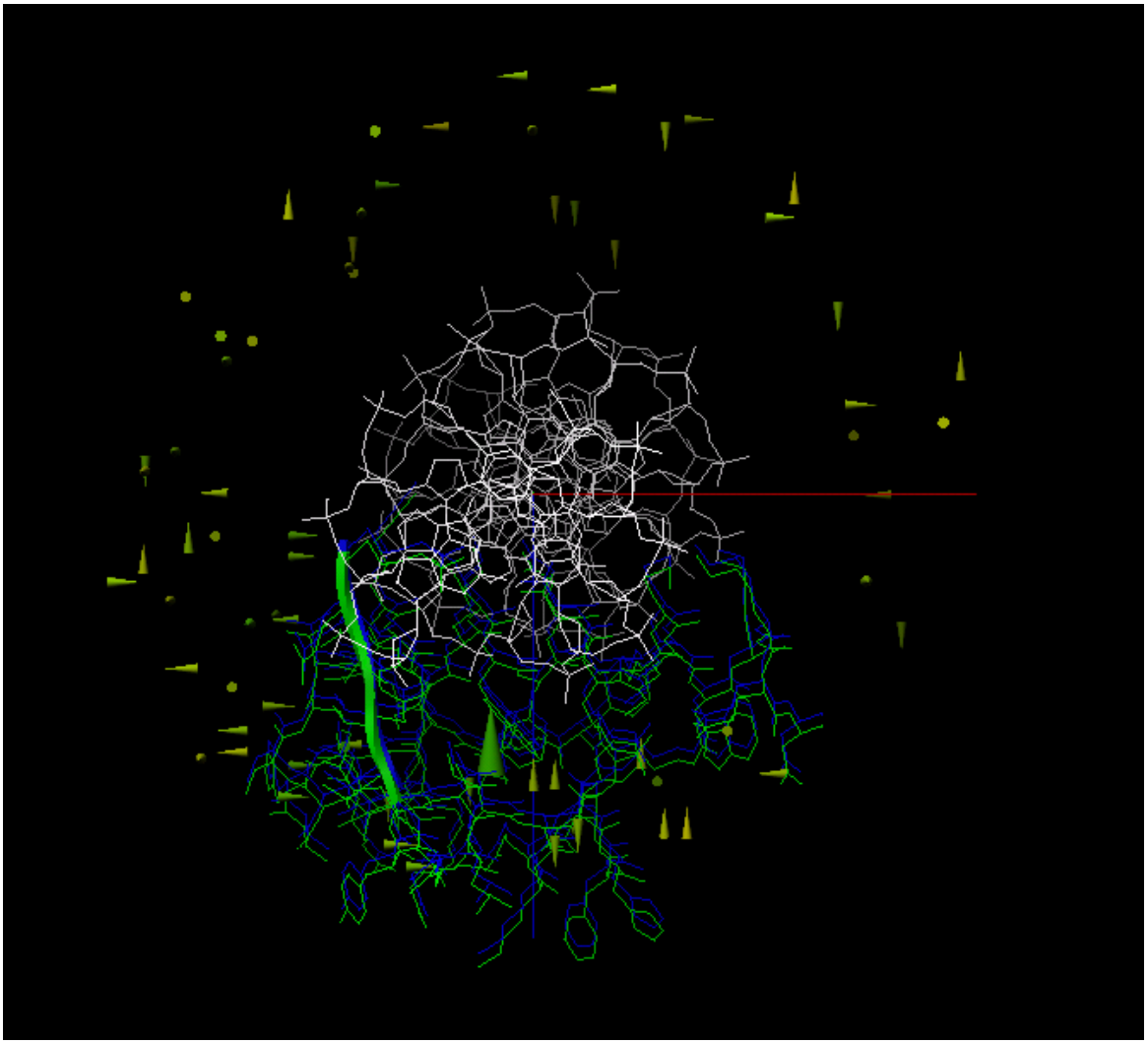

Fig. 8. Top-down view of protein in original configuration (green), and FTDock best fit (blue), with cones marking other top 100 scores. Note the large green cone, which indicates the correct orientation. Note: this is a rather artificial test case, with angle step of 90 degrees (note the orthogonal orientation of the cones).

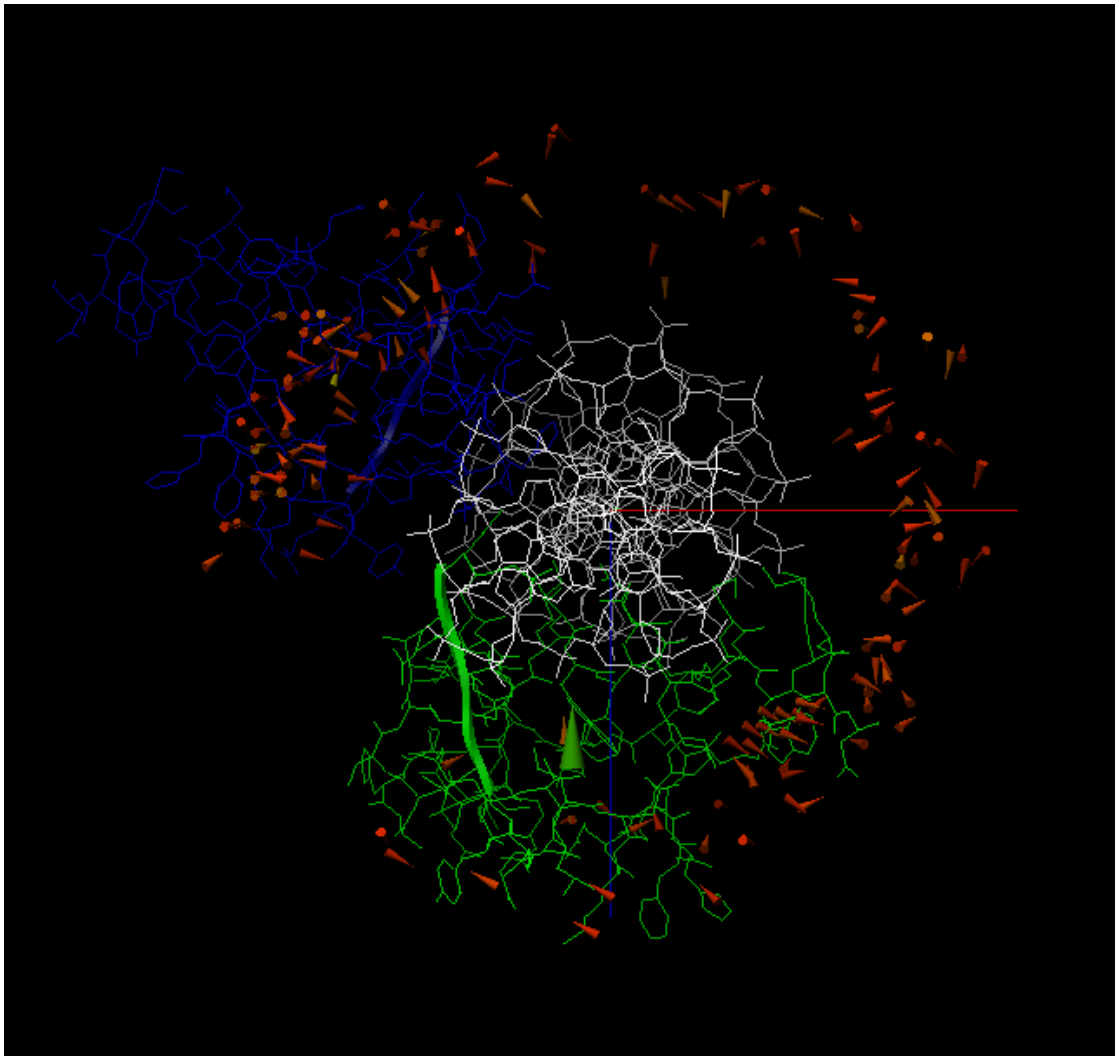

Fig. 9. A more typical result. For gridsize 100, anglestep 2, no electrostatics.

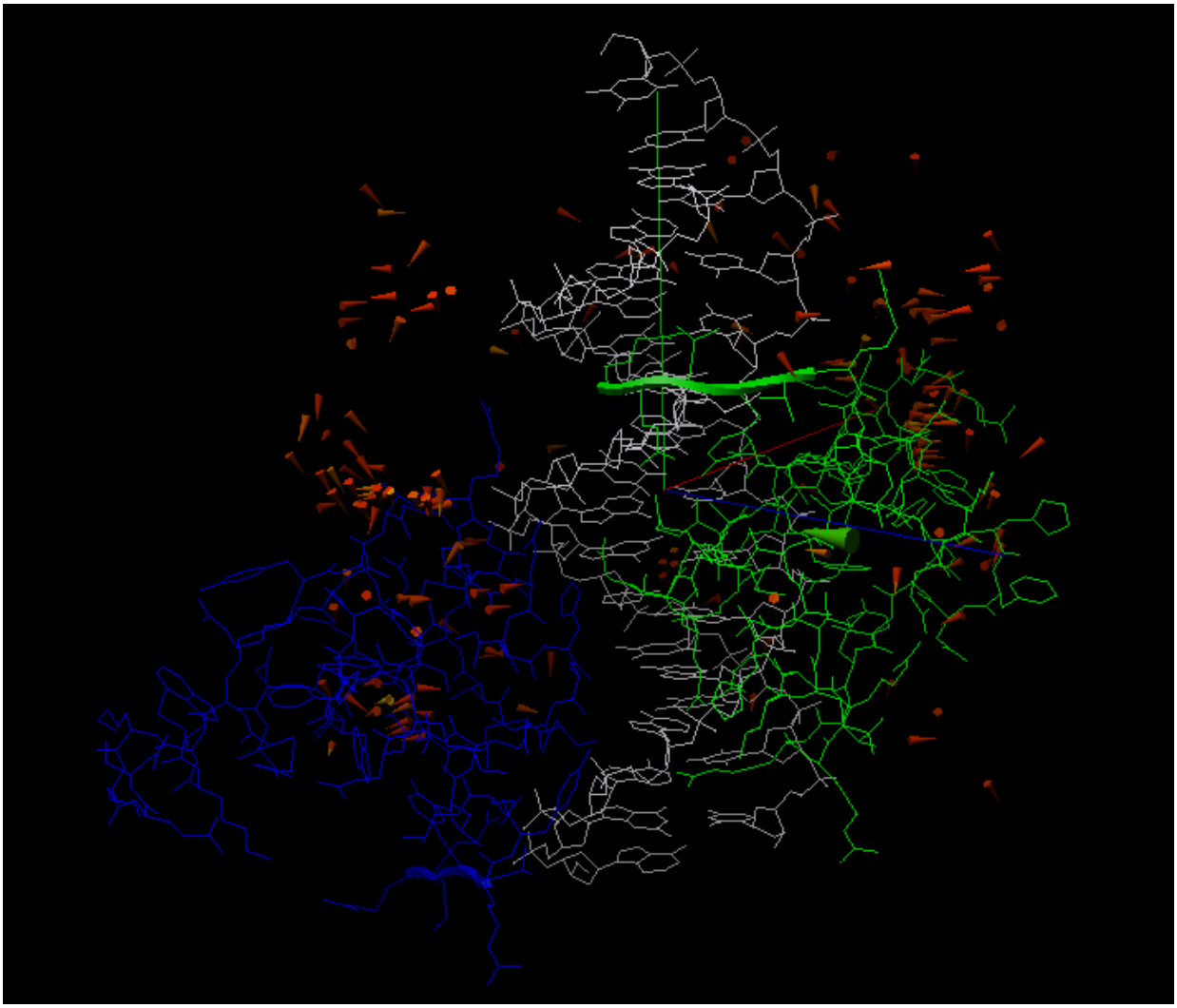

Fig. 10. Same result, sideview.

Handling DNA electrostatics

FTDock did not include code to assign charges to DNA, so had to add this to the assign_charges routine. Starting with the OPLS parameter set for DNA, averaged over each atom type, and condensed this down to: C1*, C2, C4 = 0.5; N = -0.5; O = -0.5; P = 0.8; all others set to 0.

VRML Cones

I wanted to show the FTDock results graphically, as vectors or cones surrounding the DNA molecule. Unfortunately there don't seem to be any good vector plotting modules for Chimera, Python, VMD, or InsightII. But as it turned out, Chimera can load VRML files, so I was able to convert the scores file output by FTDock to a VRML file with cones located at the center of the best configuration and oriented towards the normally DNA-facing side of the protein. The cone provides 2 angular degrees of freedom, with the 3rd missing. This could be remedied by coloring one side of the cone differently from the other. Getting this all to work involved writing code to convert the Euler angles provided by FTDock to an axis-angle representation which is used by VRML.

Results and Discussion

The results for the iterative search were rather disappointing, though there was not enough time to try iterating to higher grid resolutions.

The following results were obtained over a total run time of around 6 hours (compare with one run of 58 hours for *global* search of grid size 132, angle step 12 degrees, and no electrostatics). The RMSD value compares the correct antp configuration with the best configuration found by FTDock for that iteration.

The energy minimized versions of the molecules were not expected to produce good results, because the N-terminus of the protein is so far out of its correct configuration. This is something that would need to be modeled eventually (somehow).

static	mobile	elec?	grid size	angle step	rmsd	notes
dna	antp	no	100	60	1.09218	
				30	31.2532	flipped
				16	31.2532	
				8	38.964	
				4	34.9385	
				2	34.9385	
				1	21.636	
dna	antp	yes	100	60	29.2563	same orientation to dna, just further down
				30	29.2563	ditto
				16	23.9652	upsidedown
				8	38.2196	not good
				4	28.8666	wrong side
				2	28.8666	ditto
				1	24.4898	ditto!
dna_em	antp_em	no	100	60	19.276	facing wrong way!
				30	29.206	
				16	23.6709	
				8	23.6709	
				4	23.6709	
				2	23.6709	
				1	23.6709	
dna_em	antp_em	yes	100	60	35.9722	
				30	29.206	
				16	29.206	
				8	23.6125	
				4	29.4674	
				2	29.4674	
				1	29.4674	

Note: the protein starts in its original orientation, and since FTDock generates its initial set of angles by starting from the angles (0,0,0), the original orientation is included in the set of initial angles. This can be remedied by spinning the protein randomly, and FTDock includes a program to do this, but I wound up not doing that due to time constraints, and because very often the correct configuration of (0,0,0) wound up being quite far down the list of best scores anyway.

Conclusion

The iterative routine should explore the angular space more efficiently than a global search, and is able to drill down to higher resolution (1 degrees) than FTDock's limitation of 5 degrees. Unfortunately I was unable to get good results, at least without iterating to higher spatial resolutions.

I was also unable to test the program with very low resolution models. The limitation of having the largest cell size be around 1 Angstrom means that I didn't really get to test the low resolution models (say of 5 Angstrom cell sizes) like I had initially planned. But this could be remedied fairly easily by modifying the FTDock code to use floating point calculations instead of integers. The code would run slower but the gain by starting with lower resolution models might be enough to make up for that.

Overall, this project shows the difficulty of finding the correct binding configuration for a protein and DNA, even when starting from the exact structure of a final known complex (to say nothing of the more difficult problem of conformational flexibility). Future work should concentrate on getting the iterative search working for the known protein and DNA structures.

Acknowledgements

Thanks to Inder Jalli, Ian Rees, Hien Tran, and Dr. Briggs for help and advice.

References

[1] E. Fraenkel, C. O. Pabo, Comparison of X-ray and NMR structures for the Antennapedia homeodomain-DNA complex. *Nature Structural Biology* **5**, 692-697 (1998).

[2] G. Moont, H. A. Gabb, M. J. E. Sternberg, Use of Pair Potentials Across Protein Interfaces in Screening Predicted Docked Complexes, Proteins: Structure, Function, and Genetics, 35, 364-373 (1999).