Visualization of a Computationally Derived Fentanyl Binding Protein

Adam Colton, Team Mole

April 21, 2022

Introduction

In recent years developments in large scale molecular simulations have allowed scientists to design novel proteins for a specific targeted use. Many of these hypothesized proteins are available for free use on the protein data bank, as pdbx/mmCIF format. I looked at a paper which used computational methods to design a sequence for a fentanyl binding protein [3]. Importantly, the 3D structure of the experimentally obtained proteins atoms were determined by X-Ray crystallography, withe the data being freely available. The paper contains supplementary plots of the binding cavity of Fent49*. These visualizations are Ligplots [2]; Although Ligplots contain lots of semantic detail important to biochemists, they are ugly. For this project I wanted to visualize the transition of the protein structure from it's apo (non bound) to bound state. Instead of focusing on the biochemical interactions conveyed by the Ligplot, I wanted to examine the 3D structure of the protein's tertiary structure. My goals were to obtain a visually pleasing, ray traced, and animated video of the protein's transition.

PyMOL

PyMOL is a molecular visualization software ubiquitous in biochemistry visualizations [1]. I used PyMOL gui in conjunction with PyMOL python scripting to generate all of my visualizations.

Climber

Climber simulates the non linear morphing of one protein form to another [4]. I used Climber to interpolate between the apo and the bound state. For small conformational changes, prediction models like Climber that minimize interresidue distances are good at predicting intermediate structures.

Project Justification

On its own this project will not advance the field of science. But I think that the quality and polish of molecular visualization software such as PyMOL can only be improved when people are frequently using the software for rendering complicated visualizations. Even though I won't be able to approach the level of chemical expression demonstrated by the visualizations in the Computational Design paper, I believe that realistic looking lighting and shading rendered into a smooth video has its purposes. Biological molecules in reality do not look realistic and shaded as they exists at a scale where visible light has a larger wavelength than the small resolution of molecules. That being said, realistic renderings of molecular data gives our monkey brains a better grasp of the semantic 3D content of the data. If the software for producing these renders was more streamlined, maybe the Computational Design paper would have also included flashy visualizations, bringing the results of the paper to a wider audience, even if the flashy visuals don't provide much more raw bonding information than the superimposed ball and stick visualizations and Ligplots that were used.

Steps Used to Produce Visualization

First, I downloaded and extracted Climber from SimTK. I then set some environment variables.

```
$ export CLIMBERDIR=<~/where/climber/was/extracted/>
```

From a new directory containing my pdb flies, `5tvv_orig.pdb` and `5tzo_orig.pdb`, I ran these commands to isolate the `ATOM` lines, and then create an alignment:

```
$ awk '/ATOM/ {print}' < 5tvv_orig.pdb > 5tvv.pdb
$ awk '/ATOM/ {print}' < 5tzo_orig.pdb > 5tzo.pdb
$ ${CLIMBERDIR}/sh/morphx.sh 5tvv 5tzo 500
```

I chose 500 as the number of steps from the Climber paper's discussion on how higher step sizes increase accuracy. After thirty minutes when the run finished, I had the output folder from the run of the 1,500 intermediate structures. Climber automatically increased the number of intermediate structures it outputted, as it did not achieve it's goal energy minimization in 500.

I wrote a file `import_files.py`, which serves to import all of the Climber outputted intermediate structures.

```
def import_files(dir:str):
import os
files = os.listdir(dir)
files.sort()
```

```
for file in files:
if file.endswith(".pdb"):
cmd.load(dir + "/" + file, "mov")
cmd.extend("import_files", import_files)
```

I then started PyMOL, and imported my script. I then ran the script to import the pdb data into the enviroment.

```
>run import_files.py
>import files 5tvv_5tzo_500step/
```

I then tried some different molecule color schemes. I found the best visualization was to display the protein structure as spheres and lines, coloring the carbons as grey. I represented the near active site residues as licorice sticks, with the carbons colored green. The near active site residues I labeled using their one letter codes. I did not use PyMOL's surface rendering because it produced extremely intense flickering when rendering videos.

I imported a fentanyl molecule from the protein databank. Fentanyl was not included in the .pdb files generated by Climber, so I had to position the fentanyl molecule by hand. Since fentanyl can exist in many different conformation shapes, I had to also adjust the bond angles by hand in order to match the fentanyl shown bound in Figure 3 of the Computational Design paper.

I then set the position of the fentanyl to be inside the active site on the frame where the protein is shown bound. PyMOL automatically interpolates the position, so the movie showed the fentanyl being positioned into the active site.

Running these PyMOL commands outputted the rendered png images for each frame of the movie.

```
> mclear
> mset 1 -120 -2
> set cache_frames=0
> mpng protein-out-mov
```

I then ran this ffmpeg command to merge the images into a video. It took about 45 minutes for the rendering to complete, on my desktop PC running on my 3700x CPU.

```
ffmpeg -framerate 30
-i protein-out-mov%04d.png
-s:v 1280x720
-c:v libx264
-crf 18
-pix_fmt yuv420p
-preset veryslow
```

```
-tune stillimage
-r 30
output_vid.mp4
```

For other static images, such as the ones included in this report, I used surface rendering of the protein. PyMOL uses an algorithm to predict solvent accessible sections of the protein in order to determine where its 'surface' is. In order to be able to see the active site proteins, I did not include them as part of the surface.

Results

My movie animation can be downloaded from my website, along with the PyMOL file here. The animation is much better at conveying movement than the still images. The fentanyl carbons are colored pink, the mostly non interacting protein residues have grey carbons, and the Fen49* active site residues are colored green.

Figure 8 shows an interesting mechanism the protein uses, you can also see it in the animation. The active site S35 is at the end of a large levering arm section of the protein that moves together. Also, the overall volume of the binding site isn't obviously decreased from apo to bound. From this visualization, you can see how the protein clamps down along one axis of the ligand. Levering this S35 into the correct position is one of the energetic steps to overcome in the binding to the fentanyl molecule. The serine's carboxyl group strongly interacts with the fentanyl amide. Seeing how this protein moves between the two states highlights the chemistry involved in ligand binding.

I found I118 to be a residues not included in the paper's Ligplot, which undergoes one of the most significant translational changes upon bonding. Figures 5 and 6 show this action. Upon binding to the fentanyl, this isoleucine sticks out it's hydrocarbon side chain to position it near the center of the fentanyl's phenyl ring. If I had to guess, the aromatic phenyl ring is probably interacting with the C-H bonds, the electron density being transferred from the pi bonds to the hydrocarbon side chain.

In the bound structure, residue F9 is adjacent to the fentanyl's other phenyl ring. This is shown in the still images in Figures 9 and 10. The interaction between these two rings is a good example of pi stacking, and it probably produces a good amount of the attractive force between the fentanyl and the protein. However, the F9 phenyl ring actually widens upon binding, when maybe you'd expect it to clamp down on the fentanyl.

Effectiveness

I found that PyMOL was suitable for my needs, but possesses many undesirable traits. Working with many frames can turn surface generation into a 6 hour painfully single threaded schlep. PyMOL's gui interaction is confusing, but the command interpreter worked for most tasks. I was disappointed with the flickering in PyMOL's surface rendering when generating videos. This is probably because the surface algorithm does not correctly utilize frame by frame details to produce a temporally coherent surface. The surface algorithm must include some temporal information, as it runs completely single threaded when rendering multiple frames.

In the static images in this report, I used the computed protein surface as I felt like it adds more depth to the visualization. Even when using ray traced shadows, it can be hard to make out the 3D structure from the sea of atom spheres. The rendered video could make up for this discrepancy by including camera motion. For still images, the sphere renderings do not give our brains enough information to make out depth in the protein. Spherical protein rendering can get crowded with huge proteins, but for this smaller example I don't think it reduced the intelligibility of the video visualization.

Climber does not do a good job of simulating the residues on the outside of the protein. During binding the active site should transition somewhat predictably and reasonably from the apo to bound state, the residues on the outside of the protein will probably be moving quickly as a result of interactions with solvent molecules. In a fully fledged molecular simulation, the effects of the solvent molecules could be included in elucidating the transition structure of the protein. A fully fledged simulation also would simulate interactions between the protein and the fentanyl ligand- Climber does not do either of these for computational reasons.

I gained the most knowledge not from looking at the ray traced animation, but from using PyMOL to interactively change through camera views and scrub through the different video frames. It can be hard to recognize the conformational changes just from the video. This project convinced me of the influence of real time interactivity on the rate of knowledge gain. Staring at a static image, or even watching an animation can't compete with interactive systems. Even being able to interactively flip my report figures from bound to unbound is much more informative than simply comparing them side by side. The paper I based this report on used 3D overlays to show the differences between Fen49, and Fen49*, but overlays also have the disadvantage of being messier.

I have observed some 3D visualizations of chemical mechanisms that use a spark like light source when high energy barriers are overcome. I feel like that would be an interesting way to convey the important energetic steps in a reaction, but I didn't have any spacial energy data available to me for this data. Another improvement that would improved my visualizations would be to include water molecules in the interaction with Fen49* binding. Water is an integral part of how this mechanism occurs, but since Climber only works with protein residues, I could not include the solvent molecules in my PyMOL project.

References

[1] *PyMol.* URL: https://pymol.org/.

[2] Andrew Wallace and Roman Laskowski. *LIGPLOT v.4.5.3.* URL: https://www.ebi.ac.uk/thornton-srv/software/LIGPLOT/.

[3] Dahlia R. Weiss and Michael Levitt. "Can morphing methods predict intermediate structures?" eng. In: *Journal of molecular biology* 385.2 (Jan. 2009). S0022-2836(08)01367-3[PII], pp. 665–674. ISSN: 1089-8638. DOI: 10.1016/j.jmb.2008.10.064. URL: https://doi.org/10.1016/j.jmb.2008.10.064.

[4] Dahlia R. Weiss and Michael Levitt. "Can morphing methods predict intermediate structures?" eng. In: *Journal of molecular biology* 385.2 (Jan. 2009). S0022-2836(08)01367-3[PII], pp. 665–674. ISSN: 1089-8638. DOI: 10.1016/j.jmb.2008.10.064. URL: https://doi.org/10.1016/j.jmb.2008.10.064.

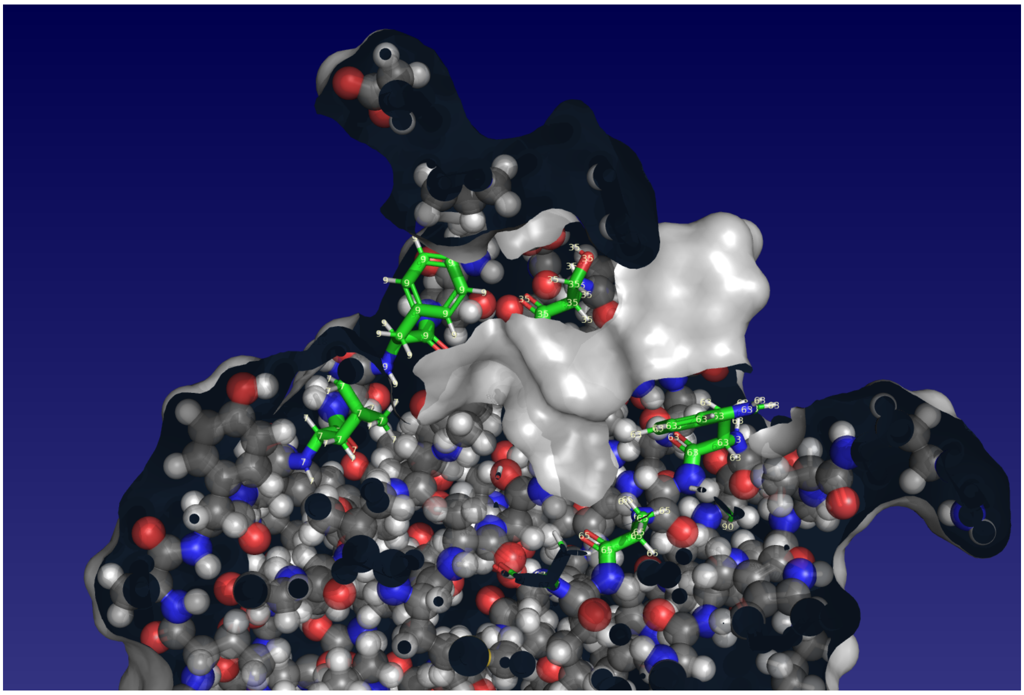

Figure 1: Cross sectional view of apo Fen49*.

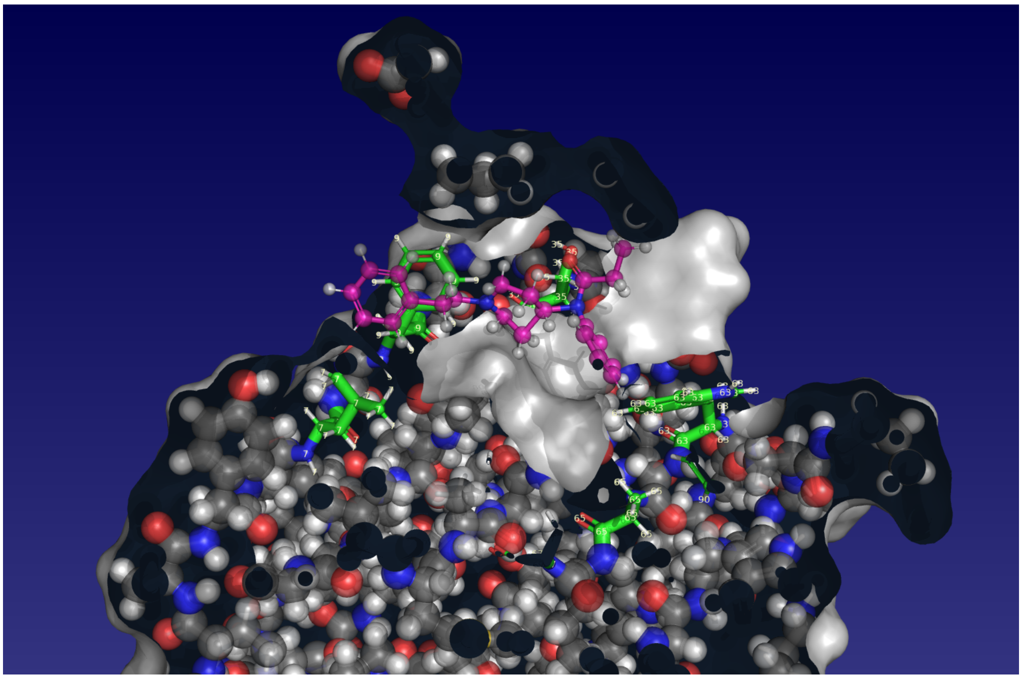

Figure 2: Cross sectional view of Fen49* with bound fentanyl.

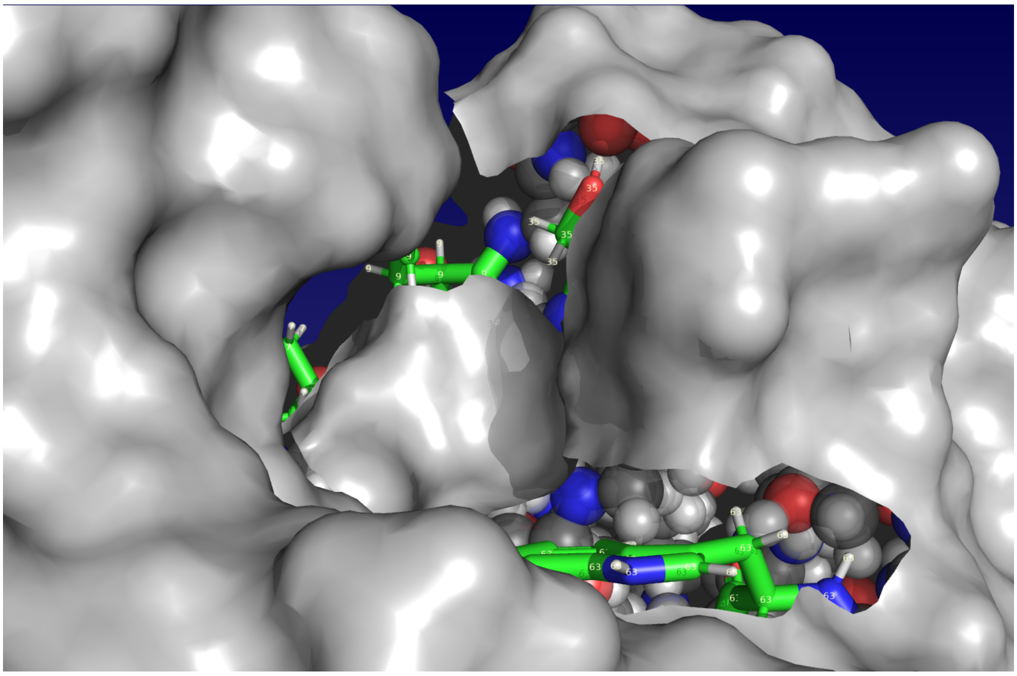

Figure 3: View of apo Fen49* binding cavity.

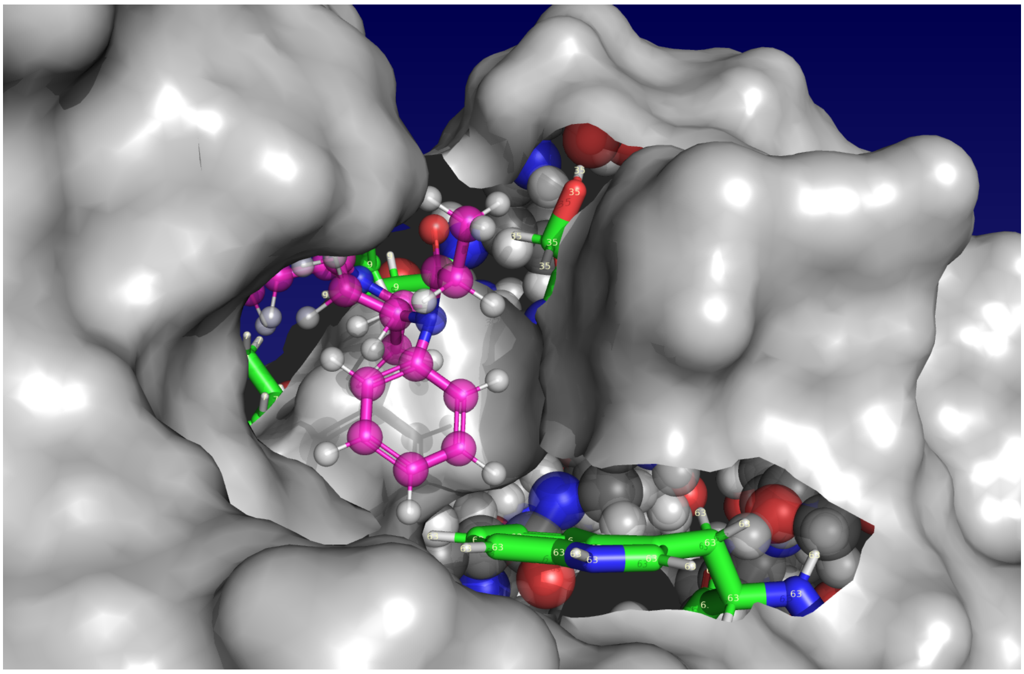

Figure 4: View of Fen49* binding cavity with bound fentanyl.

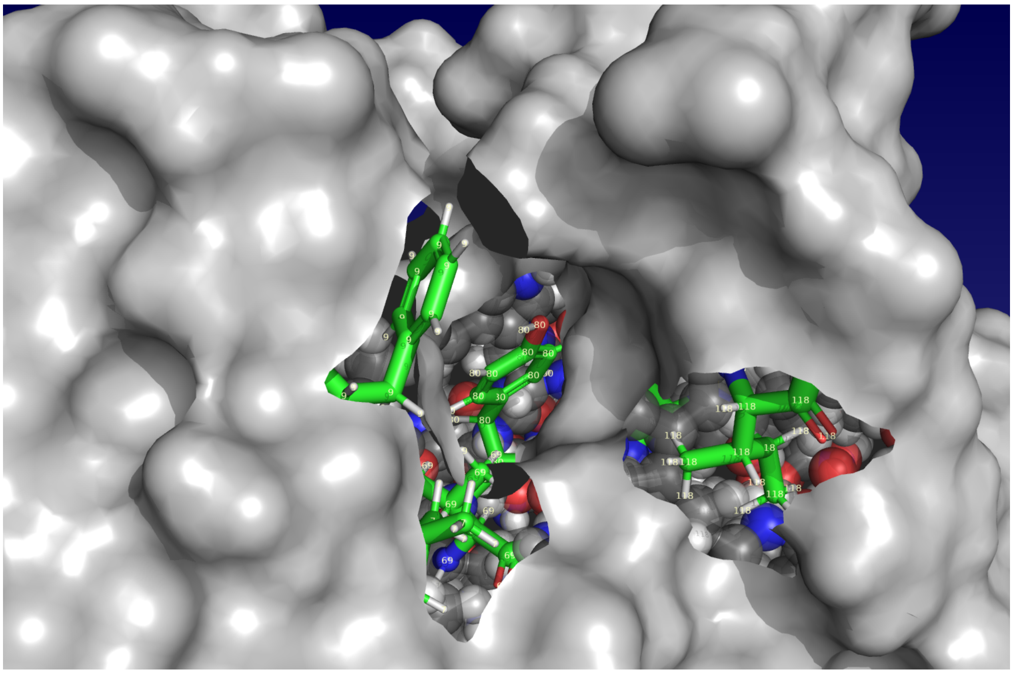

Figure 5: Cross section view of apo Fen49* highlighting residue I118.

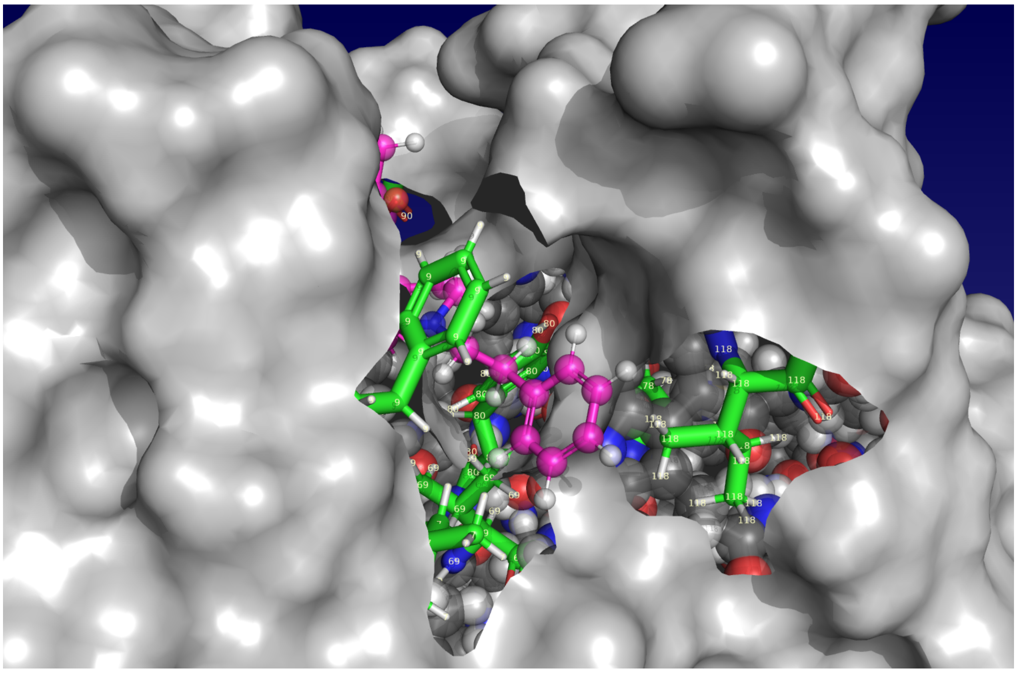

Figure 6: Cross sectional view of Fen49* binding cavity with bound fentanyl, highlighting residue I118.

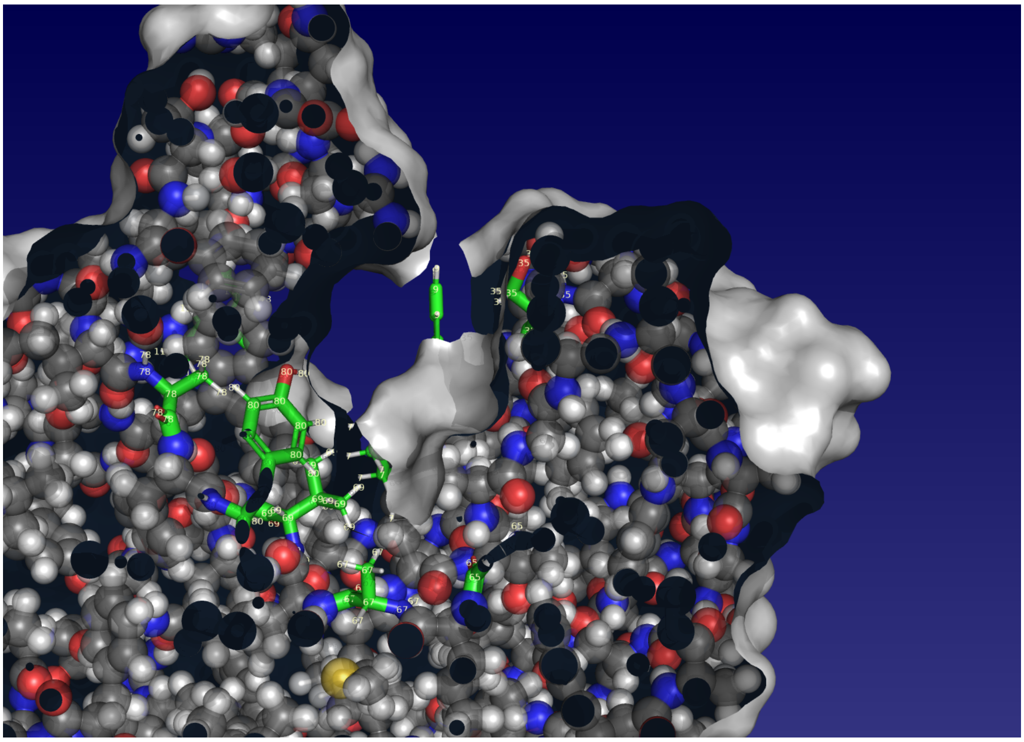

Figure 7: Cross sectional view of apo Fen49*.

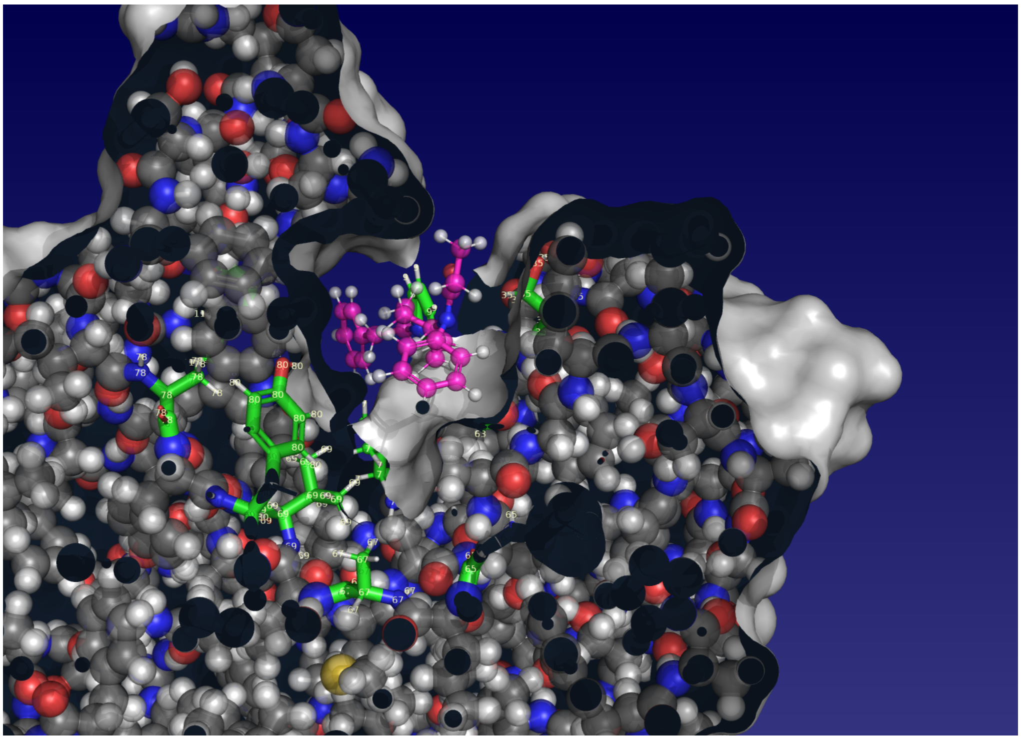

Figure 8: Cross sectional view of Fen49* with bound fentanyl. The S35 residue is at the end of a large 'levering arm' of the protein.

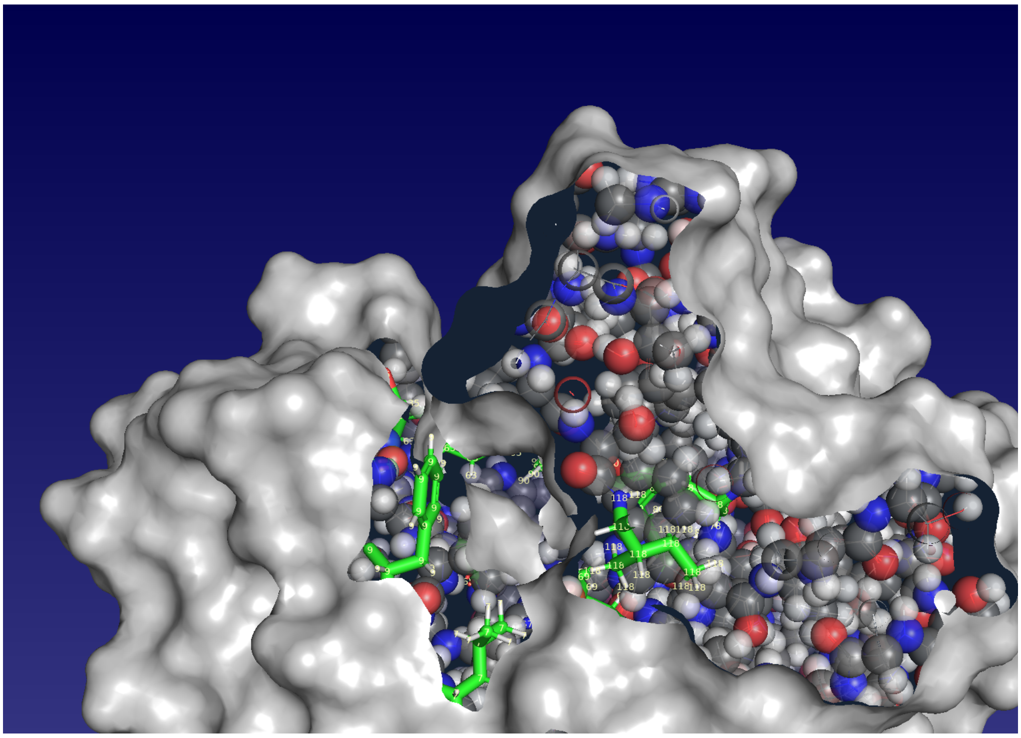

Figure 9: Cross sectional view of apo Fen49*.

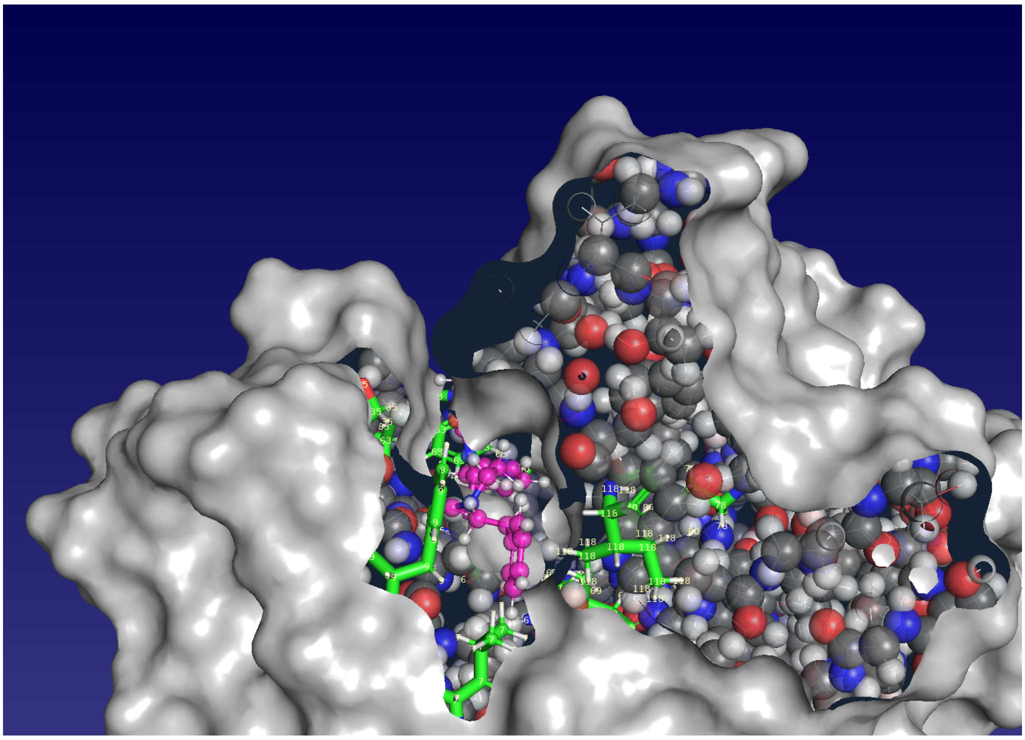

Figure 10: Cross sectional view of Fen49* with bound fentanyl. F9 undergoes slight adjustment, probably interacting with the fentanyl's phenyl ring.